WRDC-TR-90-8007
Volume VIII
Part 42

AD-A248 980

1

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 42 - Electronic Documentation System (EDS) Document Type Definition Unit Test Plan

S. Barker, F. Glandorf

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

DTIC
ELECTE
APR 21 1992
S B D

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

92-10049

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

92 4 20 079

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations

DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:

BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed form our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UTP620344903	5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR-90-8007 Vol. VIII, Part 42
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services 6b. OFFICE SYMBOL *(if applicable)*	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI
6c. ADDRESS *(City, State, and ZIP Code)* 2970 Presidential Drive Fairborn, OH 45324-6209	7b. ADDRESS *(City, State, and ZIP Code)* WPAFB, OH 45433-6533
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF 8b. OFFICE SYMBOL *(if applicable)* WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464
8c. ADDRESS *(City, State, and ZIP Code)* Wright-Patterson AFB, Ohio 45433-6533	10. SOURCE OF FUNDING NOS.
11. TITLE See block 19	PROGRAM ELEMENT NO. 78011F; PROJECT NO. 595600; TASK NO. F95600; WORK UNIT NO. 20950607
12. PERSONAL AUTHOR(S) Structural Dynamics Research Corporation: Barker, S., Glandorf, F., et al.	
13a. TYPE OF REPORT Final Report 13b. TIME COVERED 4/1/87-12/31/90	14. DATE OF REPORT *(Yr., Mo., Day)* 1990 September 30 15. PAGE COUNT 60
16. SUPPLEMENTARY NOTATION WRDC/MTI Project Priority 6203	

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify block no.)*
FIELD	GROUP	SUB GR.	
1308	0905		

19. ABSTRACT *(Continue on reverse if necessary and identify block number)*

This unit test plan establishes the methodology and procedure to be used to test the capabilities of the Electronic Documentation System (EDS) Document Type Definition computer program.

BLOCK 11:

INTEGRATED INFORMATION SUPPORT SYSTEM
Vol VIII -User Interface Subsystem

Part 42 - Electronic Documentation System (EDS) Document Type Definition Unit Test Plan

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS	Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b. TELEPHONE NO. *(Include Area Code)* (513) 255-7371	22c. OFFICE SYMBOL WRDC/MTI

EDITION OF 1 JAN 73 IS OBSOLETE

DD FORM 1473, 83 APR

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

SUBCONTRACTOR	ROLE
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface,and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

Figures

SECTION 1

GENERAL

1.1 Purpose

This Unit Test Plan (UTP) establishes the methodology and procedures used to adequately test the capabilities of the computer program identified as the EDS Document Type Definition Builder (DTDBLD). The Document Type Definition Builder is one configuration item of the Integrated Information Support System (IISS) Electronic Documentation System (EDS).

1.2 Project References

[1] Systran, ICAM Documentation Standards , IDS150120000C, 5 September 1983.

[2] International Organization for Standardization, Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML) , ISO 8879, 15 October 1986.

[3] International Organization for Standardization, Office Document Architecture/Office Document Interchange Format , ISO/DP 8613/1-6, October 1985 (Draft).

[4] American National Standards Institute, American National Standard for Information Systems - Computer Graphics - Metafile for the Storage and Transfer of Picture Description Information , ANSI X/3.122-1986, August 27, 1986.

[5] Structural Dynamics Research Corporation, Form Processor User's Manual , UM 620244200A, 16 February 1987.

[6] Structural Dynamics Research Corporation, Virtual Terminal Operator Guide , OM 620244000A, 16 February 1987.

[7] M.E. Lesk, LEX - Lexical Analyzer Generator, IS Workbench

for VAX/VMS Programmers Guide .

[8] Structural Dynamics Research Corporation, Form Processor Development Specification , DS 620244700A, 16 February 1987

1.3 Terms and Abbreviations

American Standard Code for Information Interchange (ASCII) : The character set defined by ANSI x3.4 and used by most computer vendors.

Attribute : A characteristic used to qualify an element within a document.

Character Set : A mapping of a character repertoire onto a code set such that each character is associated with its coded representation.

Compound Document : A document which may contain mixed content (text, graphics, etc.).

Computer Graphics Metafile (CGM) : A standard file format for the storage and retrieval of picture description information.

Computer Program Configuration Item (CPCI) : An aggregation of computer programs or any of their discrete portions, which satisfies an end-use function.

Conforming SGML Application : An SGML application that requires documents to be conforming SGML documents, and whose documentation meets the requirements of this International Standard.

Context-Directed Editor : An EDS application which guides the user through the process of document creation and revision by using the document type definition as a model for which logical elements may be included in the document.

Descriptive Markup : Information added to a document that

enables an application program to process the document.

Document Type Definition (DTD) : Rules determined by an application that apply SGML to the markup of documents of a particular type. A document type definition includes a formal specification, expressed in a document type declaration, of the element types, element relationships and attributes, and references that can be represented by markup. It thereby defines the vocabulary of the markup for which SGML defines the syntax. A DTD can also include comments that describe the semantics of elements and attributes, and any application conventions.

Electronic Documentation System (EDS) : An integrated set of software tools and application programs which operate upon a document through various stages of a document life cycle consisting of editing (creating/revising), formatting, imaging, storage, and transfering.

Element : A component of the hierarchical structure defined by a document type definition; it is identified in a document instance by descriptive markup, usually a start-tag and end-tag.

Element Declaration : A markup declaration that contains the formal specification of the part of an element type definition that deals with the content and markup minimization.

Entity : A collection of characters that can be referenced as a unit.

Entity Declaration : A markup declaration that assigns an SGML name to an entity so that it can be referenced.

Entity Reference : A reference that is replaced by an entity.

Field : Two-dimensional space on a terminal screen.

Form : A structured view which may be imposed on windows or other forms. A form is composed of fields. These fields may be

defined as forms, items, or windows.

Form Definition (FD) : Form definition Language after compilation. It is read at run-time by the Form Processor.

Form Definition Language (FDL) : The language in which electronic forms are defined.

Form Editor (FE) : A subset of the IISS User Interface that is used to create definitions of forms. The FE consists of the Forms Driven Form Editor and the Forms Language Compiler.

Form Hierarchy : A graphic representation of the way in which forms, items, and windows are related to their parent form.

Form Language Compiler (FLAN) : A subset of the FE that consists of a batch process that accepts a series of form definition language statements and produces form definition files as output.

Form Processor (FP) : A subset of the IISS User Interface that consists of a set of callable execution-time routines available to an application program for form processing.

Forms Driven Form Editor (FDFE) : A subset of the FE which consists of a forms-driven application used to create Form Definition files interactively.

Generic Identifier (GI) : A name that identifies the element type of an element.

IISS Function Screen : The first screen that is displayed after logon. It allows the user to specify the function to access and the device type and device name on which to work.

Integrated Information Support System (IISS) : A test computing environment used to investigate, demonstrate, and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS

addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

Item : A non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined areas where user data may be input/output.

Layout Style : The specification of format and presentation for logical elements.

Layout Structure : The hierarchy of all layout elements (pages, frames, blocks, etc.) for a document.

Logical Structure : The hierarchy of all logical elements (paragraphs, sections, etc.) within a document.

Markup : Text that is added to the data of a document in order to convey information about it.

Markup Minimization : A feature of SGML that allows markup to be minimized by shortening or omitting tags, or shortening entity references.

Message : Descriptive text which may be returned in the standard message line on the terminal screen. Messages are used to warn of errors or provide other user information.

Message Line : A line on the terminal screen that is used to display messages.

Operating System (OS) : Software supplied with a computer which allows it to supervize its own operations and manage access to hardware facilities such as memory and peripherals.

Page : Instance of forms in windows that are created whenever a form is added to a window.

Paging and Scrolling : A method which allows a form to

contain more data than can be displayed at one time with provisions for viewing any portion of the data buffer.

Parser : An application program that determines how closely a document conforms to a document type definition which defines a specific documentation standard.

Physical Device : A hardware terminal.

Previous Cursor Position : The position of the cursor when the previous edit command was issued.

Qualified Name : The name of a form, item, or window preceded by the hierarchy path so that it is uniquely identified.

Standard Generalized Markup Language (SGML) : A language for describing document structures, consisting of descriptive markup which is added to a document to indicate where logical elements such as sections and paragraphs begin and end.

Subform : A form that is used within another form.

Tag : Descriptive markup indicating the start or end of a logical element.

Tagger : An application program which provides a mechanism for automatically tagging existing documents which have been created by word processing systems.

User Interface (UI) : IISS subsystem that controls the user's terminal and interfaces with the rest of the system. The UI consists of two major subsystems: The User Interface Development System (UIDS) and the User Interface Management System (UIMS).

User Interface Management System (UIMS) : The run-time UI. It consists of the Form Processor, Virtual Terminal, Application Interface, the User Interface Services, and the Text Editor.

User Interface Services (UIS) : A subset of the IISS User Interface that consists of a package of routines that aid users in controlling their environment. It includes message management, change password, and application definition services.

User Interface/Virtual Terminal Interface (UI/VTI) : Another name for the User Interface.

Virtual Terminal (VT) : A subset of the IISS User Interface that performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by the UI software which constitutes the virtual terminal definition. Specific terminals are then mapped against the virtual terminal software by specific software modules written for each type of real terminal supported.

Virtual Terminal Interface (VTI) : The callable interface to the VT.

Window : Dynamic area of a terminal screen on which predefined forms may be placed at run-time.

Window Manager : A facility which allows the following to be manipulated: size and location of windows, the device on which an application is running, the position of a form within a window. It is part of the Form Processor.

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

SECTION 2

DEVELOPMENT ACTIVITY

2.1 Statement of PreTest Activity

During system development, the computer programs were tested progressively. Functionality was incrementally tested, and as bugs were discovered by this testing, the software was corrected.

All pretest activity was conducted by the individual developer in a manual mode. The developer would enter inputs to the DTD Builder forms, save the Document Type Definition, and then manually inspect the DTD to make sure it correctly reflected the forms input. The developer would then execute DTDBLD again, read in the test DTD, and manually inspect the forms to make sure that the DTD was read in correctly. Any errors were noted by the developer, and corrections to the Document Type Definition Builder were then made. The DTDBLD application was then re-run insure that the program was correct.

2.2 Pretest Activity Results

The results of the pretest activity were that most of the coding errors were discovered prior to the release date.

SECTION 3

SYSTEM DESCRIPTION

3.1 System Description

The Document Type Definition Builder application program provides a high-level forms based method of creating and revising SGML Document Type Definitions. It enables users who are not experts in SGML but understand the logical structure of a particular document class to construct and maintain Document Type Definitions.

The SGML language statements contained within a DTD define the logical structure of a document class. These statements define when and how many times a logical element can occur within a document. For example, a book contains many chapters, chapters contain many paragraphs, but a book normally only contains one foreword and one table of contents.

The DTDBLD application program enables the user to describe this hierarchy of logical elements by assigning level numbers to each logical element and defining the parent child relationships between each node. In addition the number of times a logical element can occur can also be defined.

Once the user defines this hierarchy, it can be saved to an output file. The DTDBLD program translates the hierarchy defined via the forms into SGML language statements and writes them into the output file. This SGML DTD is then used by other EDS application programs (see Figure 3-1). as a source of information about the logical structure of a document class.

The block diagrams contained in Figures 3-1 and 3-2 illustrate the DTDBLD test configuration used in the Unit Test Plan.

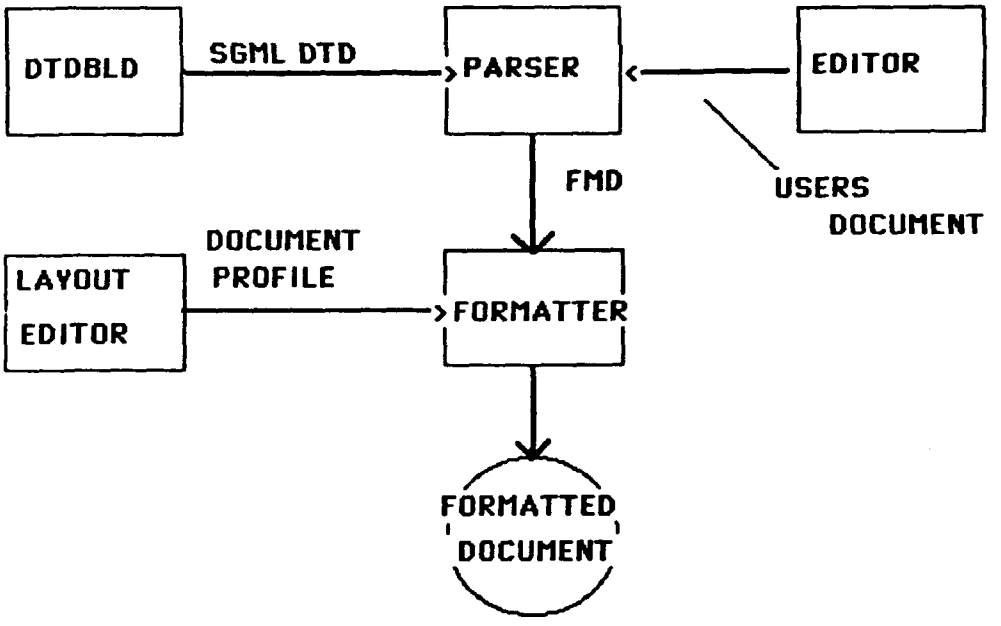

Figure 3-1 EDS Block Diagram

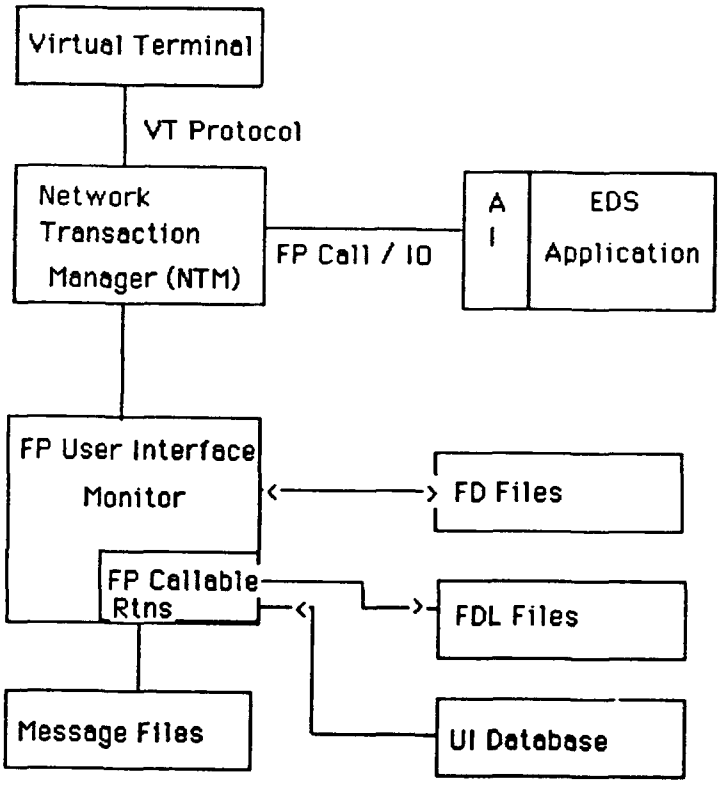

Figure 3-2 UIMS Block Diagram

3.2 Testing Schedule

Since EDS application programs use the Forms Processor (FP) and the Network Transaction Manager (NTM) subsystems, both the FP and NTM must be tested before EDS application program Unit Test Plans can be run.

3.3 First Location Testing

These tests of the Document Type Definition Builder require the following:

Equipment: Air Force VAX, terminals supported by the Virtual Te

as listed in the UI Terminal Operator's Guide.

Support Software: the Integrated Information Support System, C libraries.

Personnel: one integrator familiar with both IISS and EDS.

Training: the EDS user manual has been previously delivered.

Deliverables: the EDS Document Type Definition Builder CPCI.

Test Materials: all tests may be run interactively by inputting appropriate data and comparing the output DTD file to an existing DTD file as outlined in this test plan.
A script file has been created to run each test plan and save the resulting output for comparison in future tests.

Security Considerations: None.

3.4 Subsequent Location Testing

The requirements listed above must be met. A script file can be created and used to run subsequent executions of the Unit Test Plan.

SECTION 4

SPECIFICATIONS AND EVALUATIONS

4.1 Test Specification

The Unit Test Plan is based on covering specific functionality of the Document Type Definition Builder outlined in the Electronic Documentation System Development Specification (DS).

The following chart has the functional requirements as outlined in the EDS DS listed vertically and the test activities in the UTP that demonstrate the testing of each functional requirement listed horizontally.

	Test Codes										
Func Req.	A	B	C	D	E	F	G	H	I	J	K
Document Hierarchy Input	X										
Insert Mode		X									
Search Mode			X								
Insert After Search				X							
Delete Mode					X						
Examine Parent Child						X					
Insert Attributes							X				
Delete Attribute								X			
Save DTD									X		
Copy DTD										X	
Delete DTD											X

Figure 4-1 Matrix Mapping of Requirements with Test Plan

The following list has the test name followed by the list of figures that correspond to the test.

A Figures 5-1 to 5-7
B Figures 5-8 to 5-10
C Figures 5-11 to 5-12
D Figures 5-13 to 5-15
E Figures 5-16 to 5-16

F	Figures 5-17 to 5-21
G	Figures 5-22 to 5-24
H	Figures 5-25 to 5-28
I	Figures 5-29 to 5-31
J	Figures 5-32 to 5-34
K	Figures 5-35 to 5-36

4.2 Testing Methods and Constraints

The tests outlined in Section 5.3 must be followed in the correct order. The required input is given for each test. This testing uses the normal mode of operation and does not test every possible code path that may generate an error. It assumes that the files and logical names detailed in Section 5.2 are available when the tests are run. It also assumes a proper IISS environment is available as described in Section 5.2. During the development phase, error reporting due to missing files, incorrect logical names, and improper IISS environments was tested by the developer.

No data recording is required for the DTD Builder tests. It is suggested that upon further running of these tests, scripting of the unit test may be done and the output from running the script be saved for future testing. A script file for the DTD Builder test procedure is under Configuration Management and can be used if the tester does not want to manually key the data into the input forms.

No additional constraints are placed on this unit test besides those listed in Sections 3.3 and 5.2 of this document.

4.3 Test Progression

THE EDS DTD Builder application program is tested by creating a Document Type Definition and comparing it to an existing DTD under IISS Configuration Management. The tester will first describe a logical element hierarchy using the DTDBLD Hierarchy form, examine the parent child relationships using the

DTD Document Hierarchy Menu form, create attributes for two logical elements, then save the input to an SGML Document Type Definition. The test is successful if the output DTD matches the DTD under IISS CM.

4.4 Test Evaluation

The test results are evaulated by comparing the Document Type Definition created by the Unit Test Plan (DTDBLD.DTD) with the DTD under IISS Configuaration Management (DTDSAV.DTD). The two files can be compared using a differences command (such as VAX DIFF) and should match in every way with the exception of any instances of the date and time within the file.

SECTION 5

TEST PROCEDURES

5.1 Test Description

The Unit Test for the DTD Builder application program consists of creating a logical element hiearchy for a document class, examining the parent child relationships defined in this hierarchy, assigning attributes to two logical elements, then saving the input into an SGML DTD output file. In addition, the DTD file maintenance operations of COPY and DELELE are also tested.

The test is sucessful if the output DTD completely matches the DTD under Configuration Management. The output DTD name will be DTDBLD.DTD and will be written to the directory pointed to by the logical name EDSDTLIB. The DTD under IISS Configuration Management is DTDSAV.DTD.

5.2 Test Control

As outlined above, this unit test may be done manually or run using the supplied script files. The order of testing is completly specified in Section 5.3. The test control information is completly described by the sequence of source input forms and the output Document Type Definition.

5.3 Test Procedures

To run the unit test plan as outlined in this section on a VAX, one must be logged onto a valid IISS account. The NTM must be up and running and the UI group logical names IISSFLIB, IISSMLIB, and EDSDTLIB must be assigned correctly. IISSFLIB points to the directory containing production form defintions (FD files). IISSMLIB points to the form containing error messages (MSG files). EDSDTLIB points to the directory containing document type definitions (DTD files).

Assuming the NTM is up and running, an IISS user may start this Unit Test Plan with the supplied script file as follows:

```
$ SET DEFAULT {directory containing your NTM environment}
$ VT100 -rEDSDTD.SCP
```

This starts up the VT100 device driver with a source script as input. If the User Interface has been installed at your site with a different device driver, then this step is amended as appropriate. The tests then begin executing at the terminal.

If the user chooses to run this test manually, then the sequence of commands are as follows:

```
$ SET DEFAULT {directory containing your NTM environment}
$ VT100
```

When the IISS Logon Screen (Figure 5-1) is displayed, login to IISS with username/password/role of MORENC/STANLEY/MANAGER.

When the IISS Function Screen (Figure 5-2) is displayed, enter EDSDTD as the specified function and press <ENTER>. This will start the EDS DTD Builder application program.

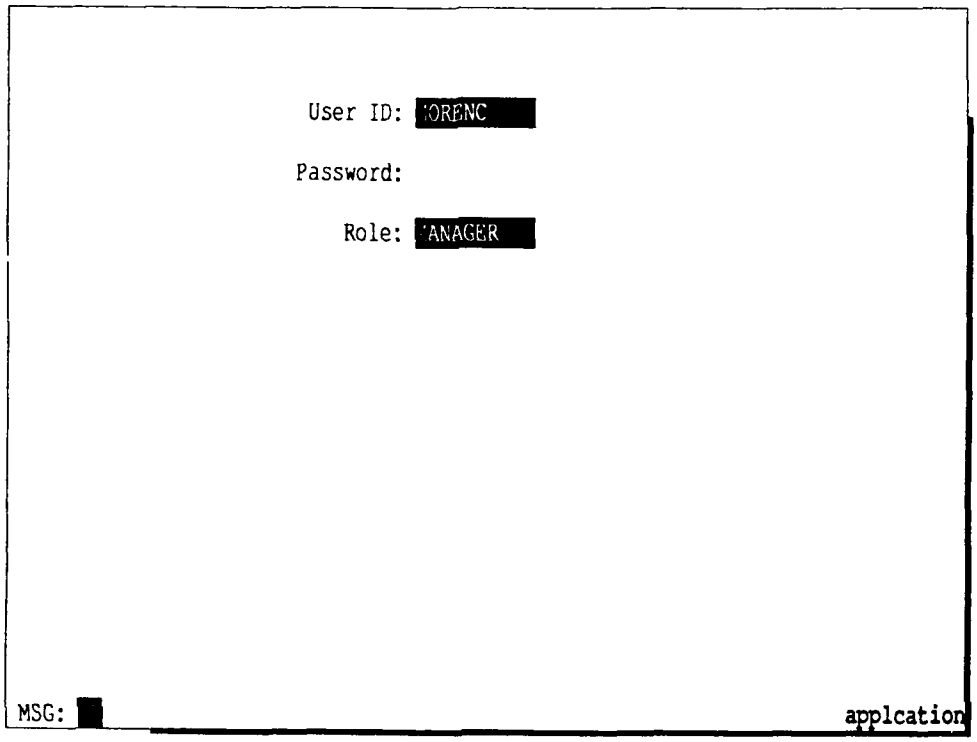

Figure 5-1 IISS Logon Screen

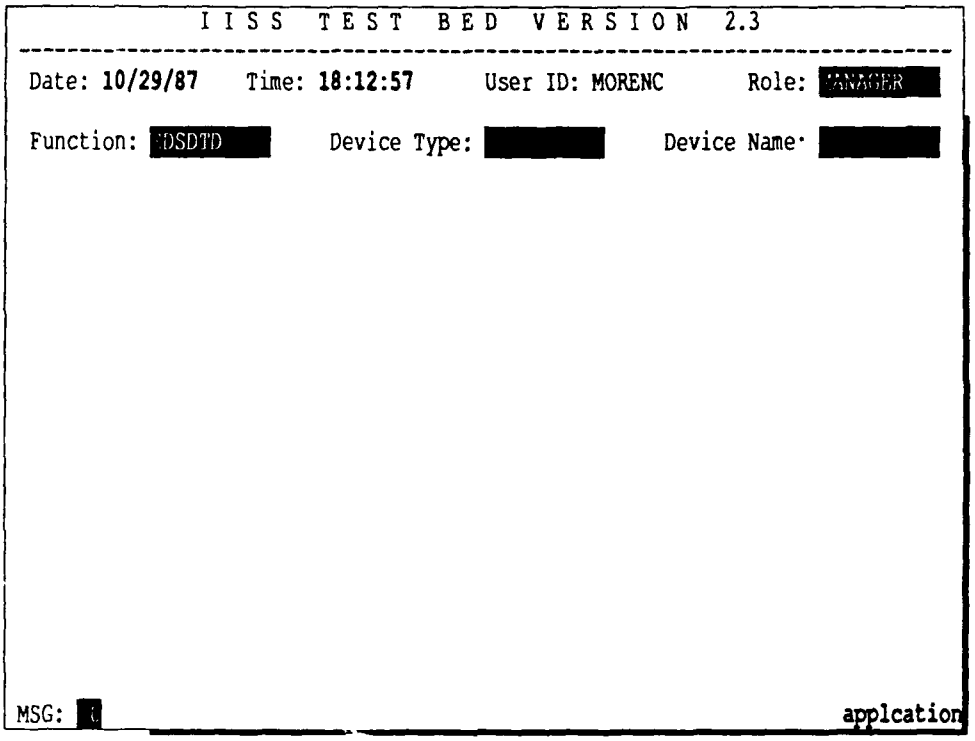

Figure 5-2 IISS Function Screen

Once the EDS DTD Builder application has started, the following form is displayed.

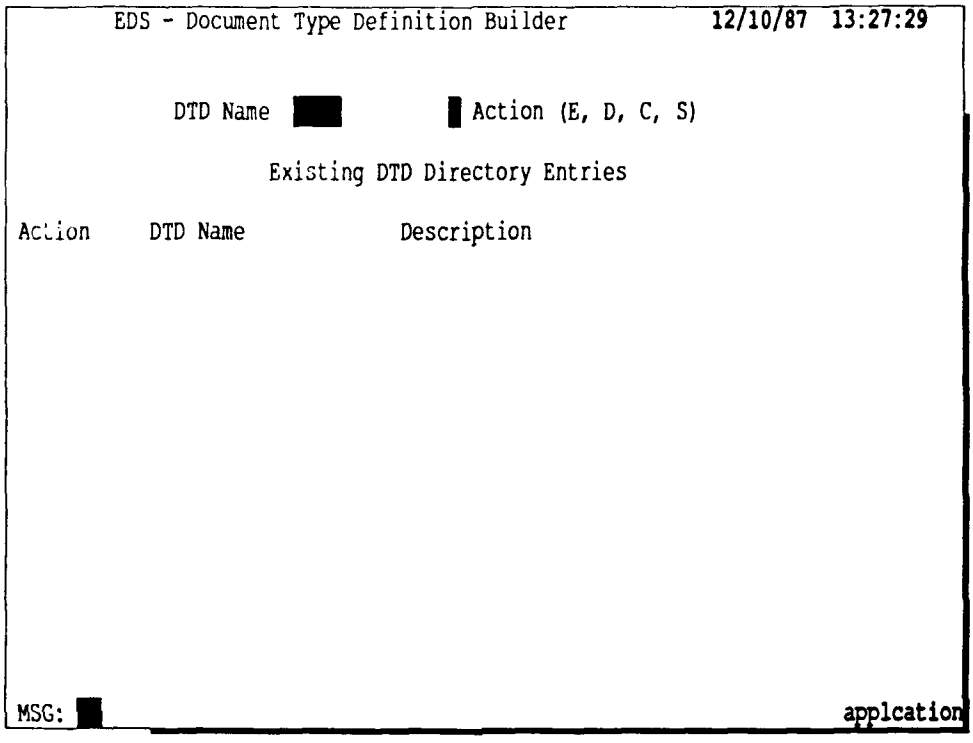

Figure 5-3 Blank Document Type Definition Form

Enter the input data shown in Figure 5-4 to create a new document type definition profile named DTDBLD, press the <ENTER> key to go to the next form.

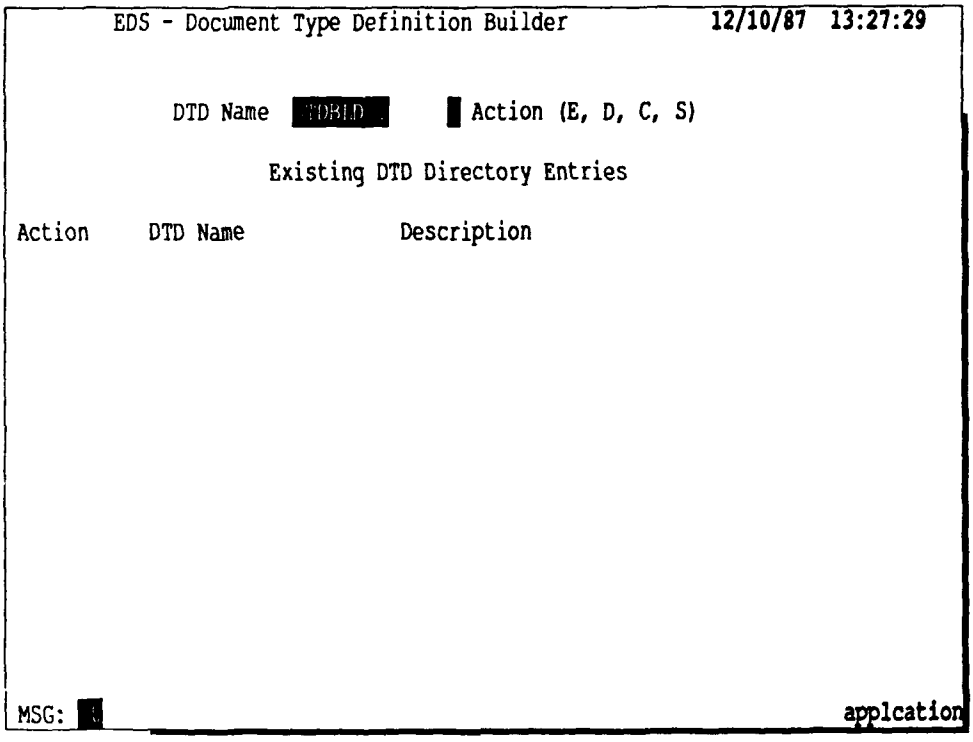

Figure 5-4 Completed Document Type Definition Form

Once the <ENTER> key as been pressed, the form shown in Figure 5-5 should be displayed and the cursor should be sitting at the Generic Identifier field at the zero level.

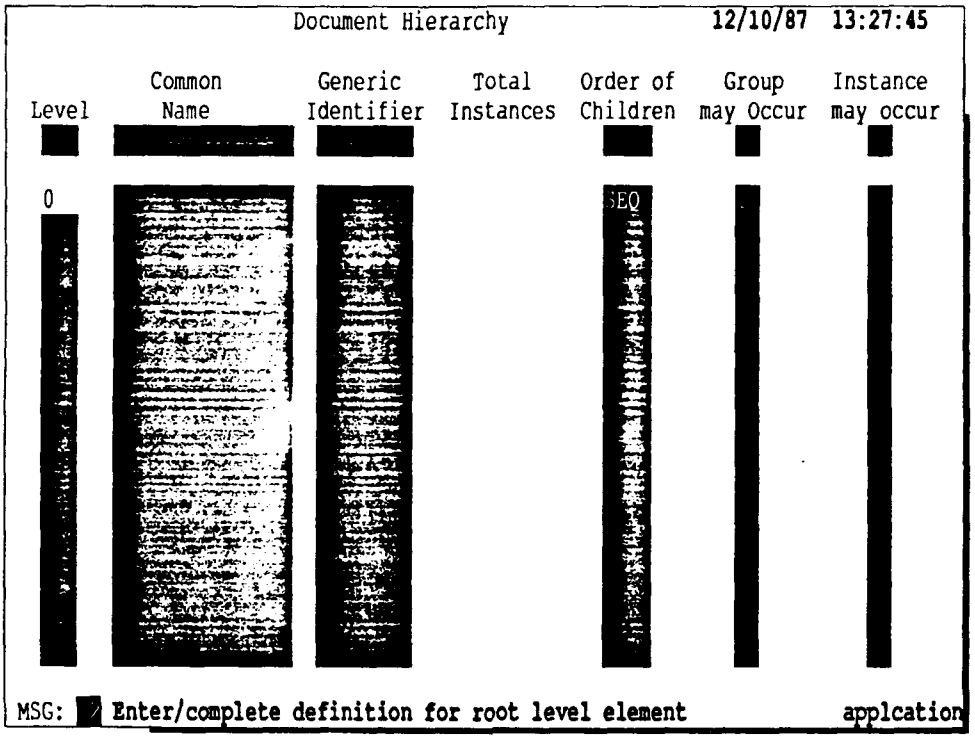

Figure 5-5 Blank Document Hierarchy Form

Enter the data shown in Figure 5-6 into the form fields.

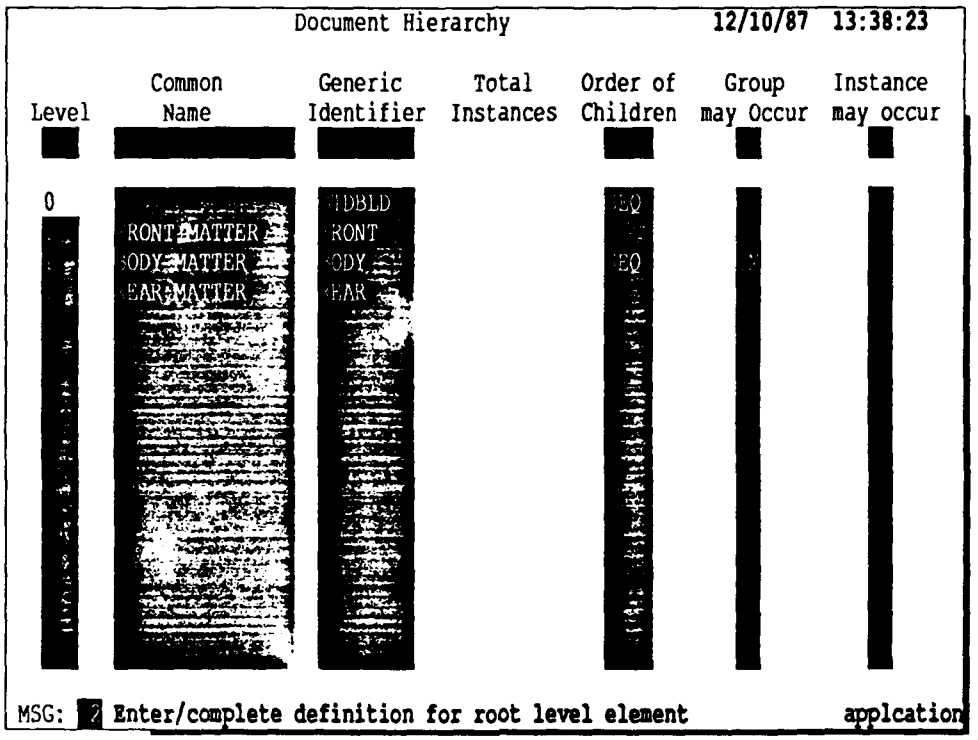

Figure 5-6 Completed Document Hierarchy Form for Test A

Once all data items have been entered, press the <ENTER> key to save the input data into the application programs internal data structures. Note that the data is not written to an output file until the <PF6> key has been pressed. After the <ENTER> key has been pressed the screen should look as shown in the following form.

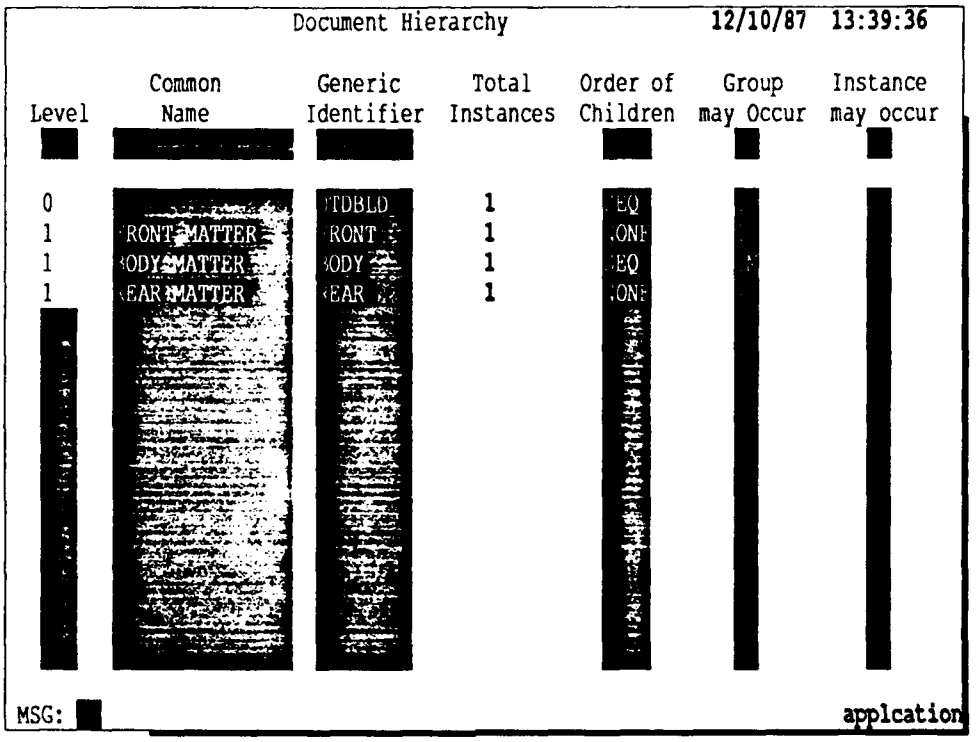

Figure 5-7 Document Hierarchy Form after Test A

Now move the cursor to the Level field of the entry that contains the Generic Identifier name BODY. To insert logical elements under BODY press the <PF5> key. The following form should be displayed.

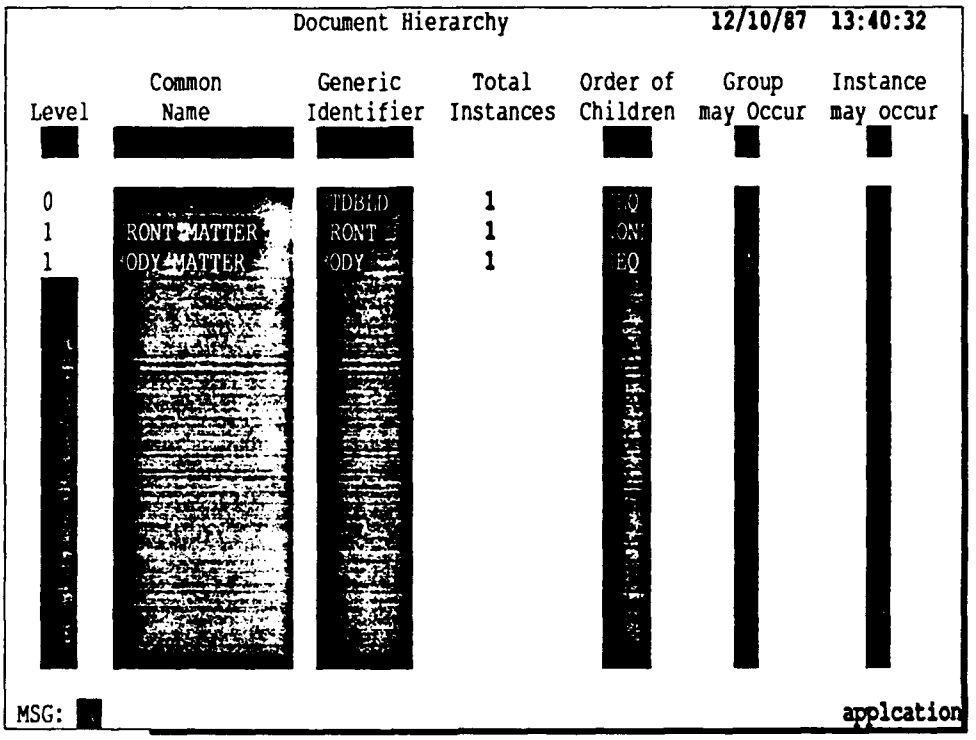

Figure 5-8 Document Hierarchy Insert Mode - Test B

The cursor should be now sitting on the blank level field under the BODY entry. Proceed to enter the data into the form fields as shown in Figure 5-9.

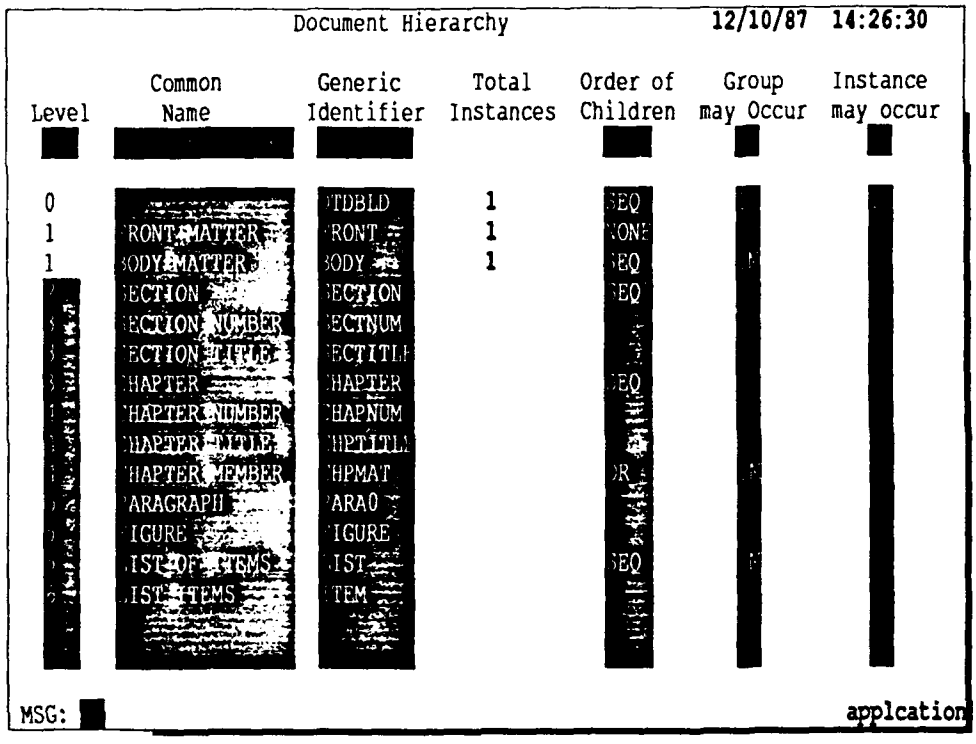

Figure 5-9 Test Data for Insert Mode Test

Once the form is complete press <ENTER> to insert the defined elements under the BODY entry. The Document Hierarchy form will again be displayed with the full hierarchy as shown in the next form. At any time the number of entries exceeds the avaiable space on the screen, scrolling and paging can be used to view additonal entries.

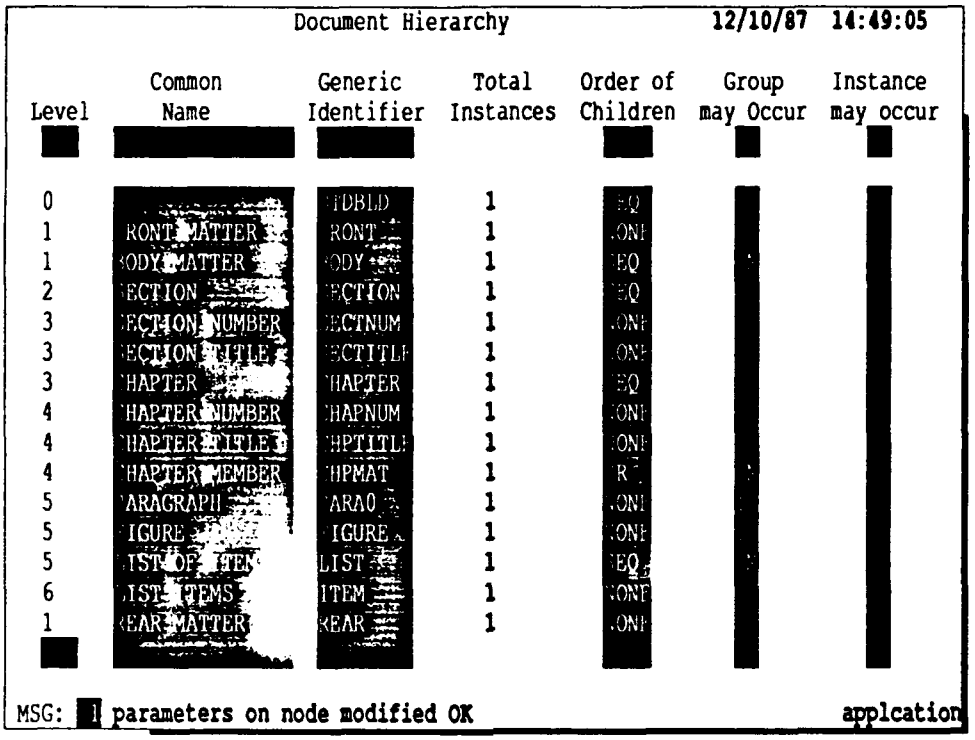

Figure 5-10 Completed Document Hierarchy Form after Test B

Now move the cursor to the Generic Identifier field of the first entry line on the Document Hierarchy form and enter the word FIGURE as shown in Figure 5-11.

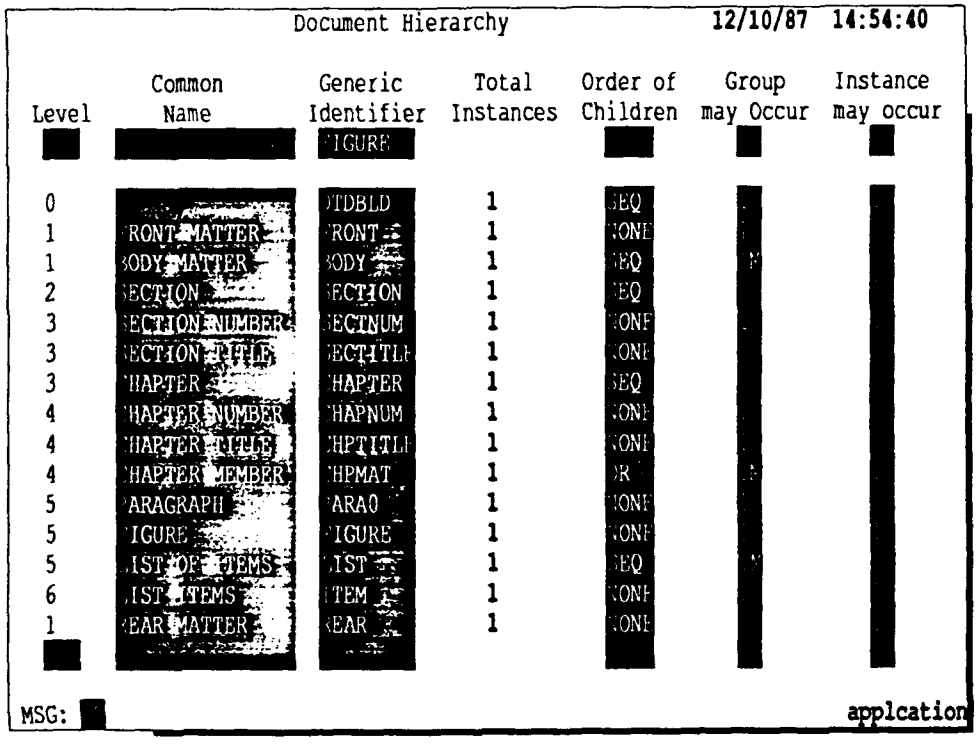

Figure 5-11 Search Mode - Test C

When the <ENTER> key is pressed, the DTDBLD application will look for the first Generic Identifer with the value of "FIGURE" and display this entry on the first input entry line in the Document Hierarchy form as shown in Figure 5-12. This enables the user to "search" the hierarchy for elements already defined and also provides a method of giving the user a full page of entries under this element in which to insert elements under as shown in Figure 5-13.

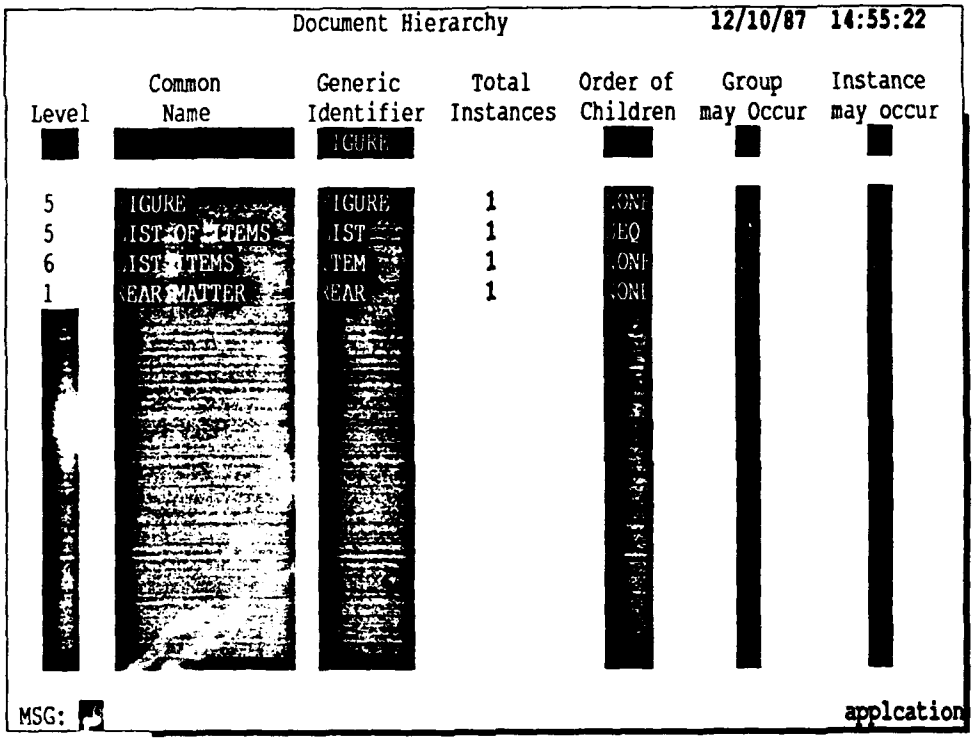

Figure 5-12 Result of Search Mode - Test C

Press the <PF8> key to erase FIGURE from the Generic Identifier field in the Search line.

Now move the cursor to the LEVEL field of the entry that contains the Generic Identifier FIGURE. Press the <PF5> key to insert elements under the FIGURE element. The following form should be displayed when the <PF5> (INSERT) key is pressed.

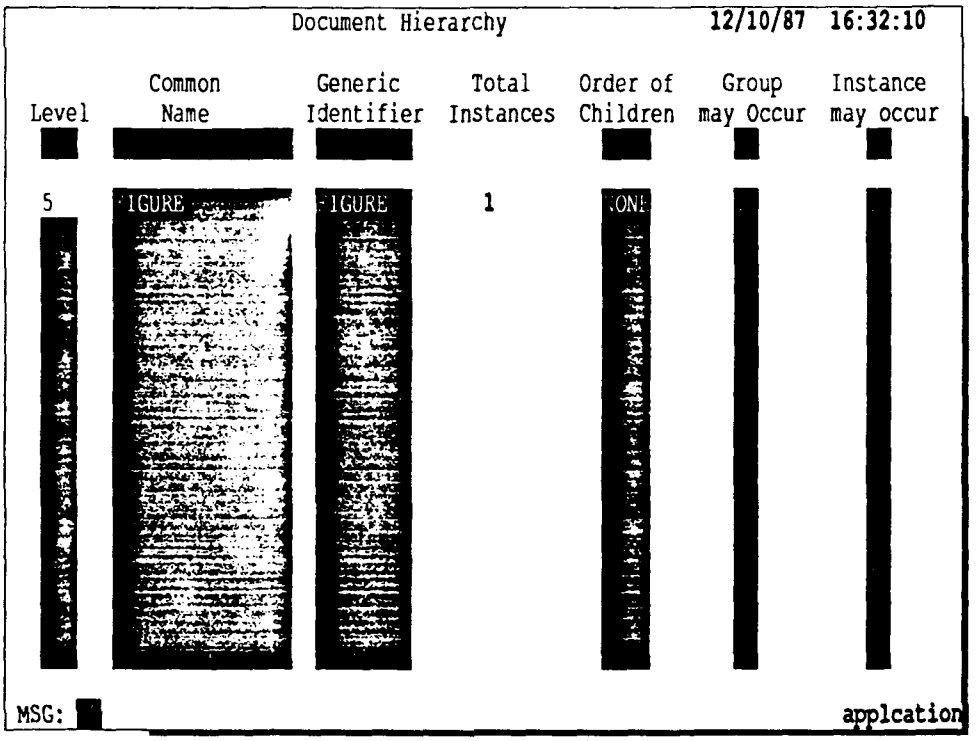

Figure 5-13 Insert after Search Mode for Element Figure - Test D

Complete the form shown in Figure 5-13 with the data in Figure 5-14. Be sure to change the ORDER OF CHILDREN field for the FIGURE Generic Identifier from NONE to SEQ to indicate that there are elements within a FIGURE.

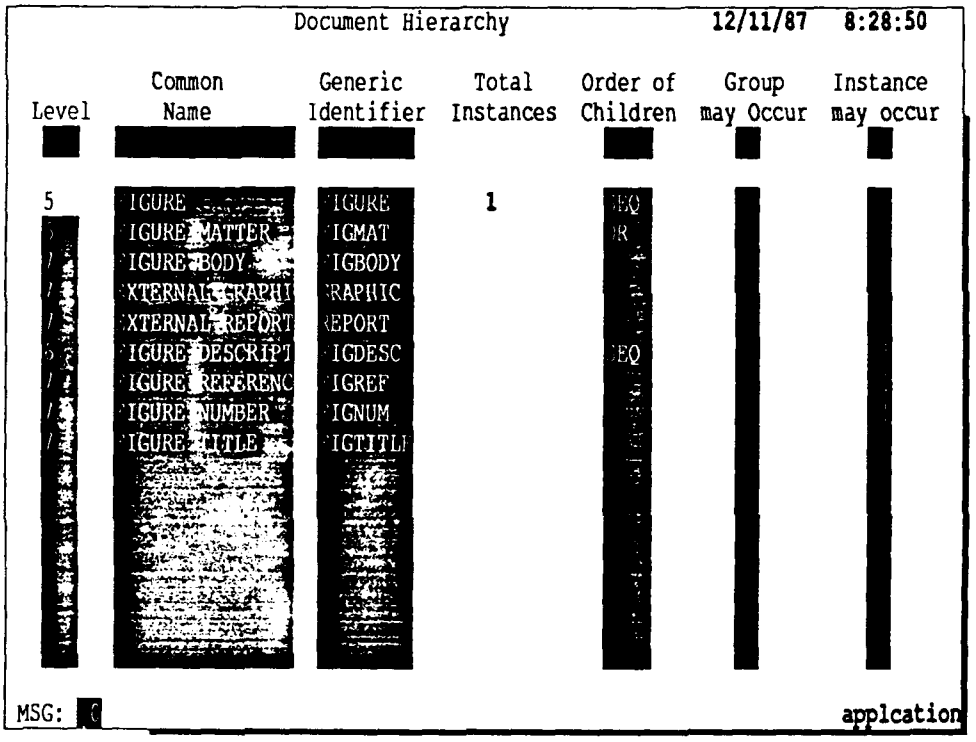

Figure 5-14 Data for Insert of Elements after Figure

After the form has been completed with the data in Figure 5-14 press <ENTER> to save the data. The DTDBLD application will leave INSERT mode and redisplay the full document hierarchy as shown in Figure 5-15.

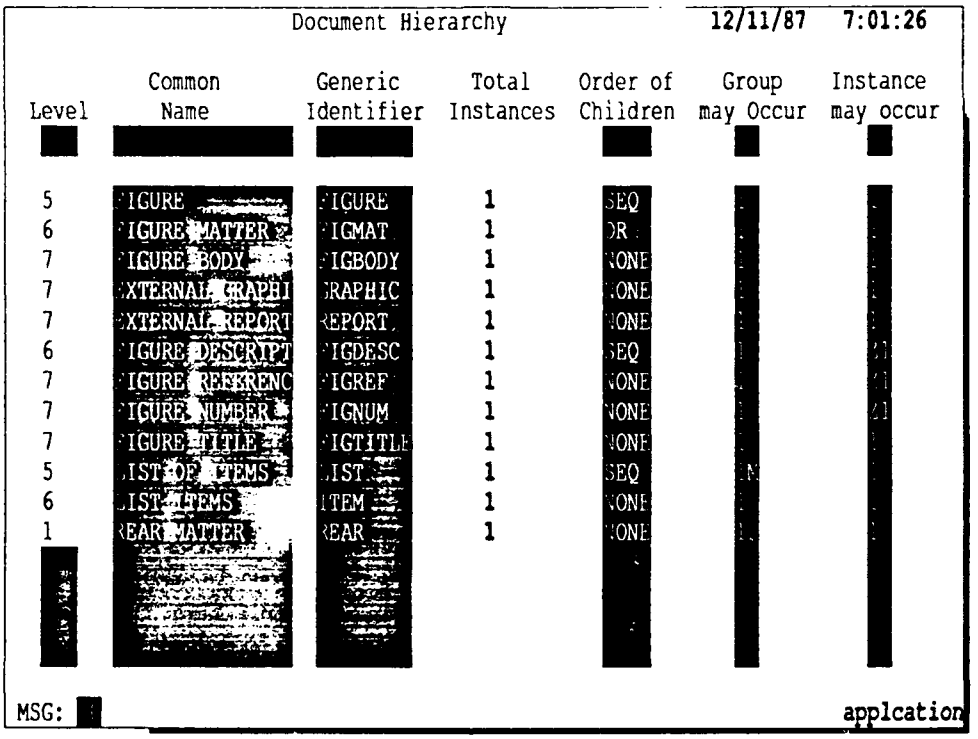

Figure 5-15 Full Document Hierarchy after Test D

To test the DELETE function, move the cursor to the LEVEL field of the entry containing the Generic Identifier REAR. Press the <PF7> key to delete this element leaving a document hierarchy form as shown in Figure 5-16.

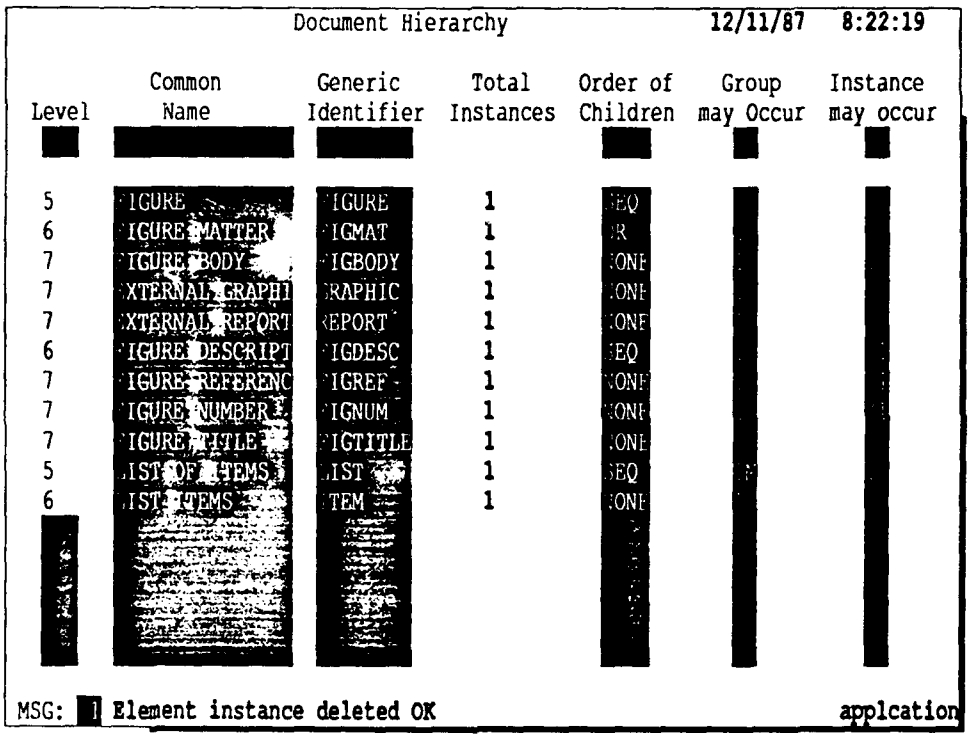

Figure 5-16 Result of Delete Function - Test E

Now move the cursor to the line containing the Generic Identifier FIGURE and press the <PF10> key (Display Hierarchy or number 5 on the keypad) key. The following form should then be displayed.

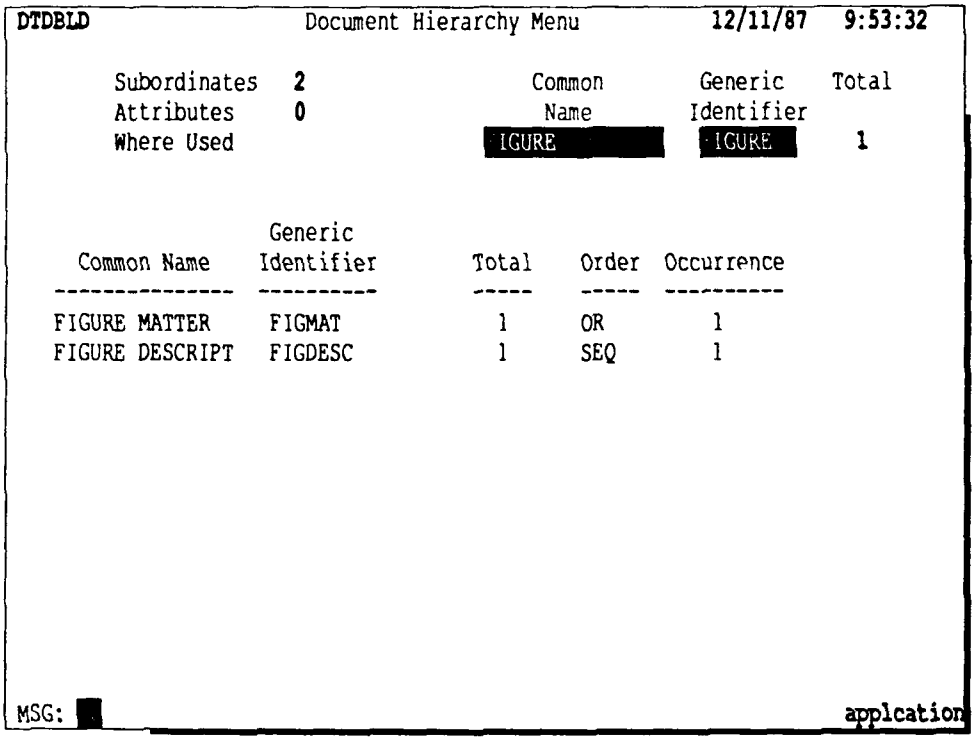

Figure 5-17 Document Hierarchy Menu Form

This form enables the user to examine the parent/child relationships in the document hierarchy. The input fields (top right) contain the name of the parent and the list of the children is presented in the form fields below.

To test this function, move the cursor using the <TAB> key to the Generic Identifier field containing the value FIGDESC. Press the <PF13> key to display the children of FIGDESC. The Document Hierarchy Menu form should then look as shown in Figure 5-18.

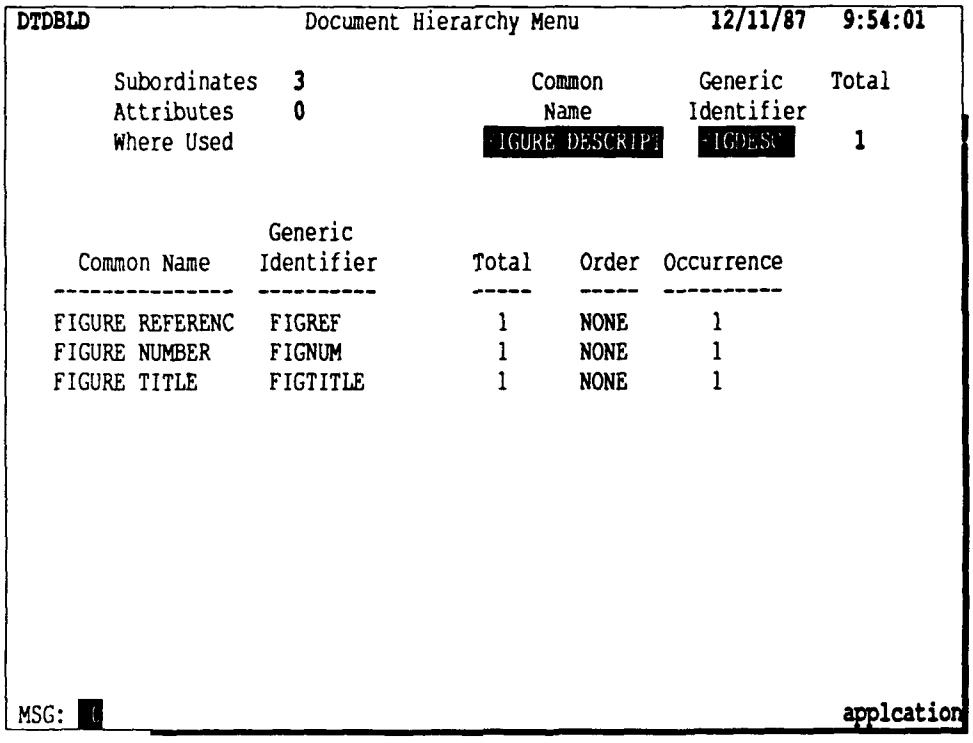

Figure 5-18 Document Hierarchy Menu Form after <PF13> key

Now move the cursor using the <TAB> key to the Generic Identifier input field containing the value FIGDESC. To display the parent of FIGDESC press the <PF14> key. The Document Hierarchy Menu form should be updated with the values shown in the next form.

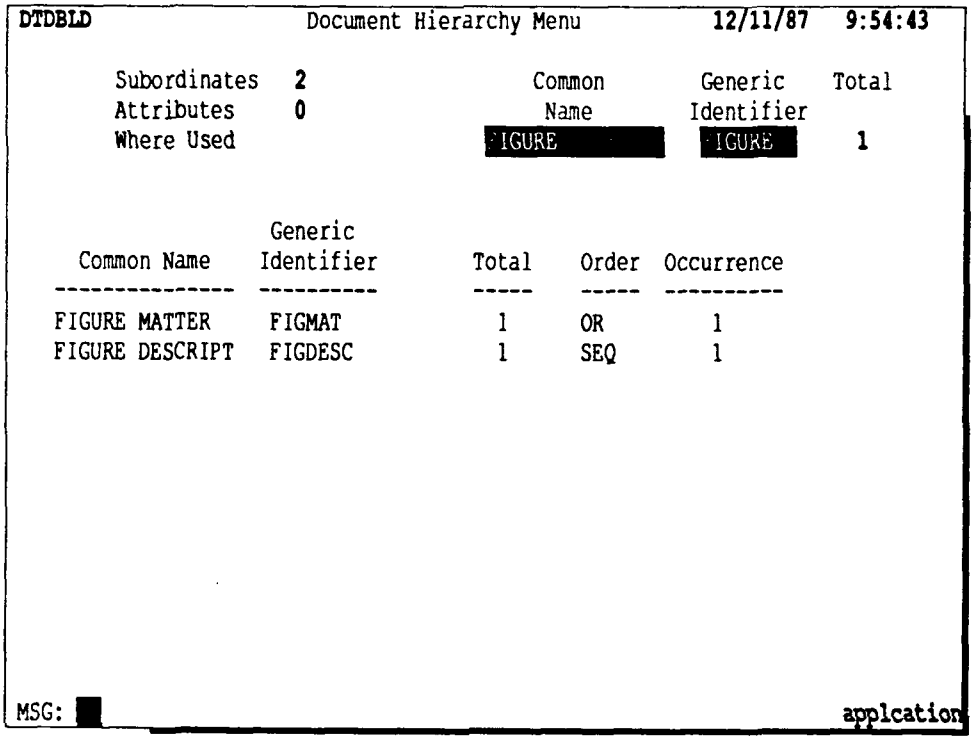

Figure 5-19 Document Hierarchy Menu Form after <PF14> Key

The next portion of the Unit Test Plan consists of assigning attributes to an element in the document hierarchy. To perform this test, first move the cursor to the Generic ID field containing the value FIGMAT and then press the <PF13> key.

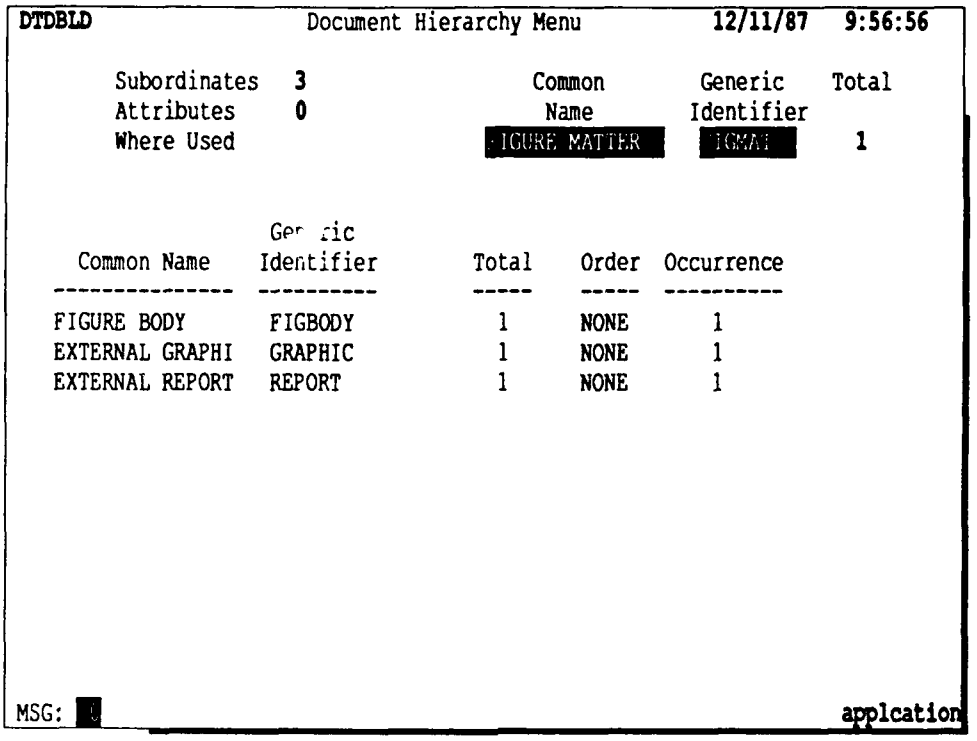

Figure 5-20 Document Hierarchy Form - Children of FIGMAT GI

Now move the cursor to the Generic Identifier field containing the value of GRAPHIC and press the <PF13> key to move the GRAPHIC element into the parent box as shown in the next Figure.

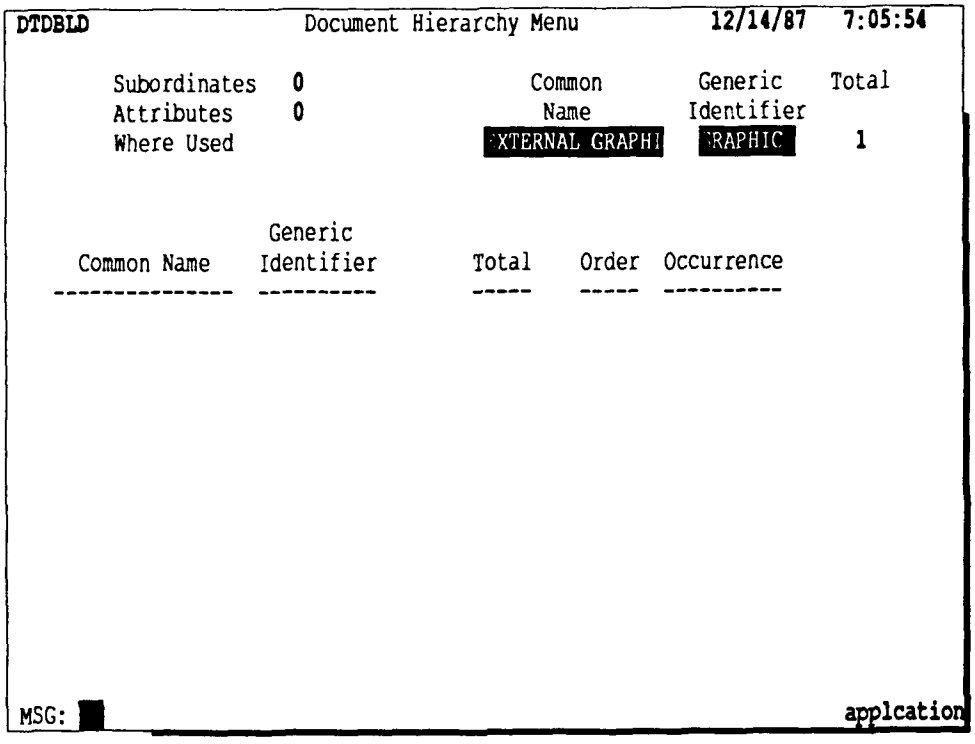

Figure 5-21 Document Hierarcy Form - Children of GRAPHIC GI

The attributes that will be assigned as part of the Unit Test Plan will be assigned to the GRAPHIC element. Now press the <PF15> (Attribute) key to proceed to the Attribute form as shown in the next Figure.

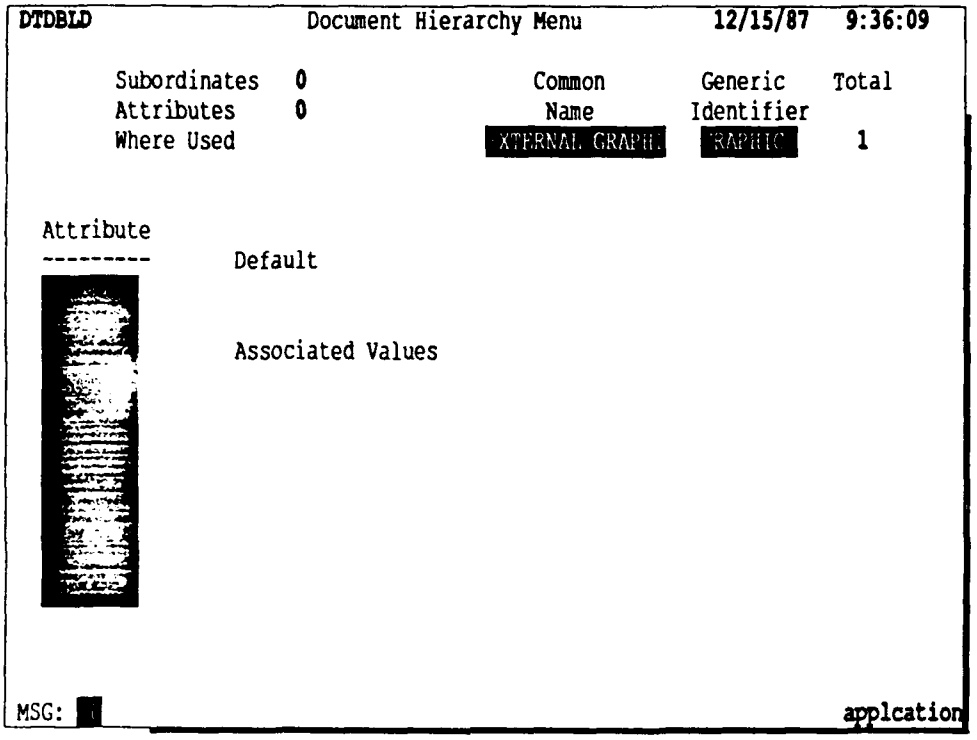

Figure 5-22 Blank Attribute Input Form

To start the Attribute Insert test, move the cursor to the first line in the Attribute input field. Enter a value of "file" into the field as shown in the next form.

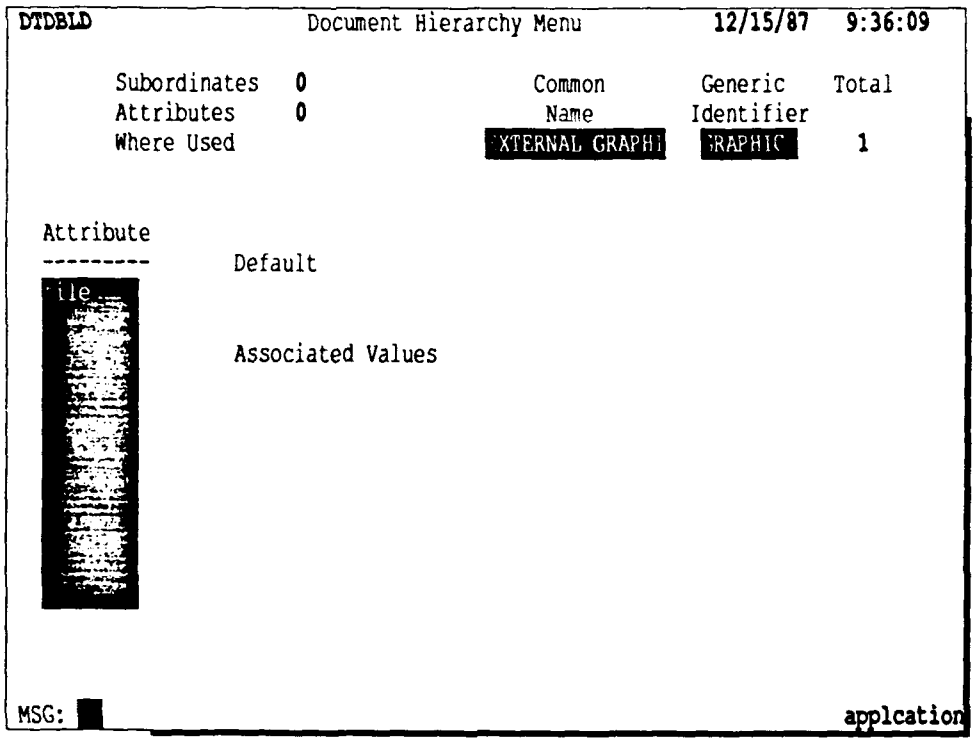

Figure 5-23 Input Data for Attribute Insert Test

Press the <ENTER> key to insert the Attribute of file into the list of attributes for the element GRAPHIC. Now move the cursor to the second Attribute input line and enter the value of "fmt". Press the <ENTER> key to select the FMT value and then proceed to enter the values for the DEFAULT and ASSOCIATED VALUES fields as shown in the next form.

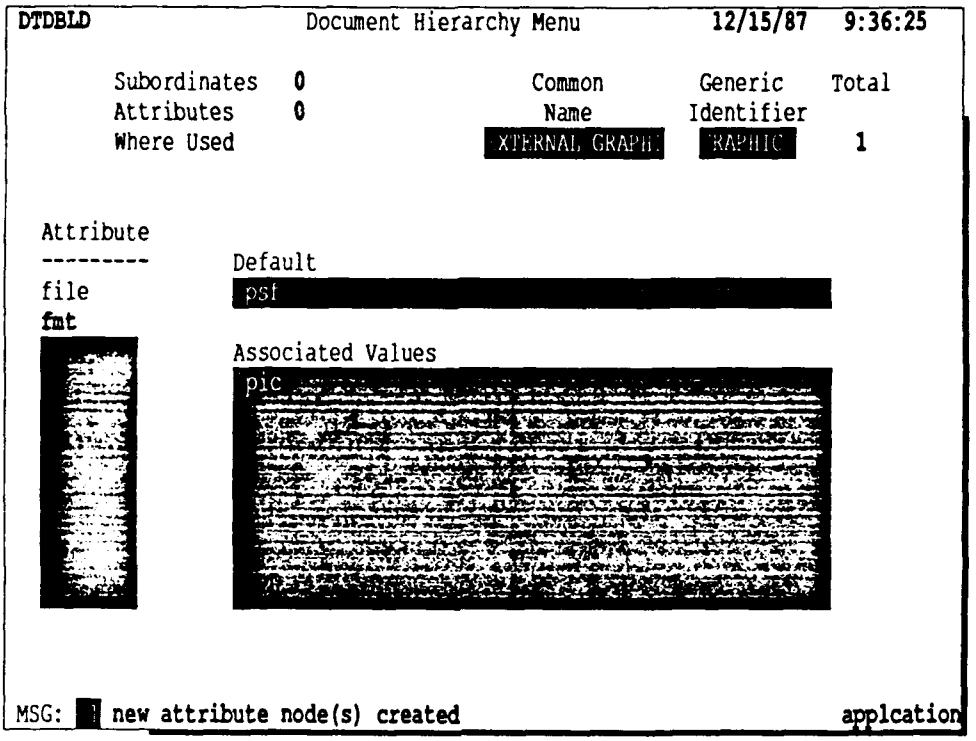

Figure 5-24 Completed Attribute Form

Press the <ENTER> key to save the new values for the fmt attribute. Now move the cursor to the third line of the Attribute list (under the value of "fmt"), enter the value "dummy", and then press the <ENTER> key. The screen should now look like the form in Figure 5-25.

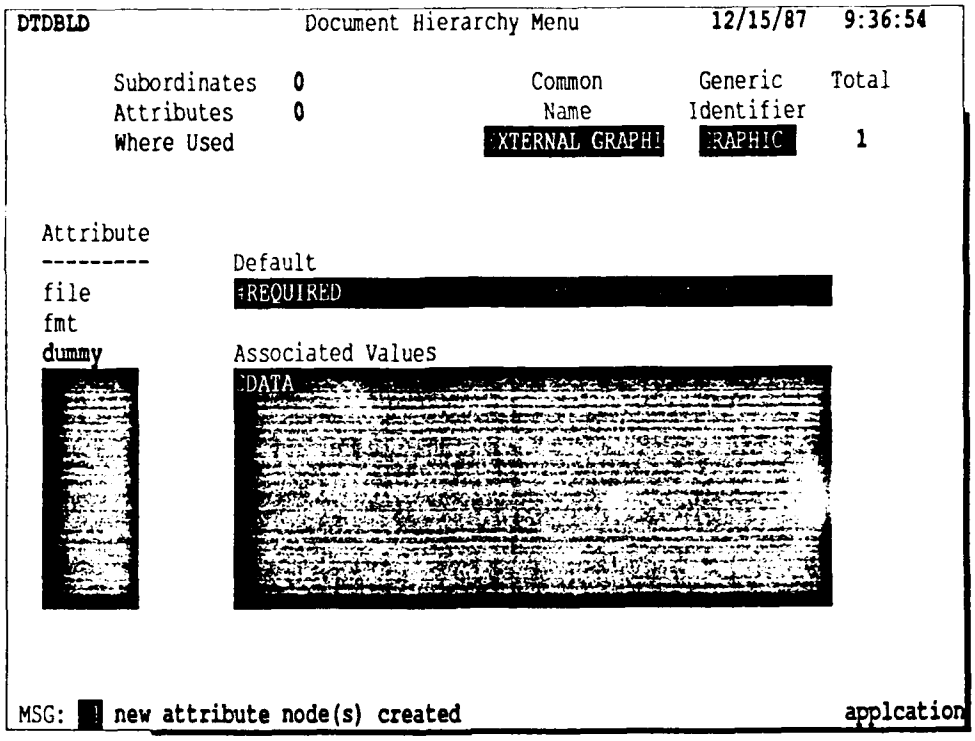

Figure 5-25 Completed Attribute Form - Setup for Delete Test

To test the delete Attribute function, move the cursor to the Attribute field containing the value of "dummy". Press the <PF7> key to delete the attribute DUMMY, leaving a form looking as follows.

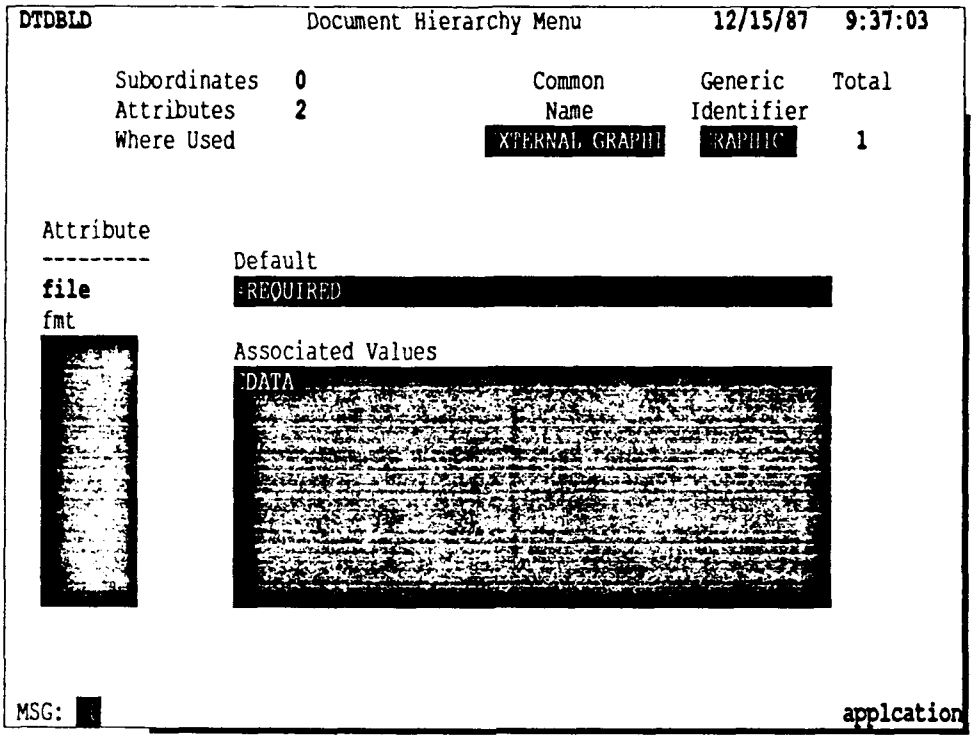

Figure 5-26 Attribute Form After Delete Test

Now press the <QUIT> key to return to the Document Hierarchy Menu form.

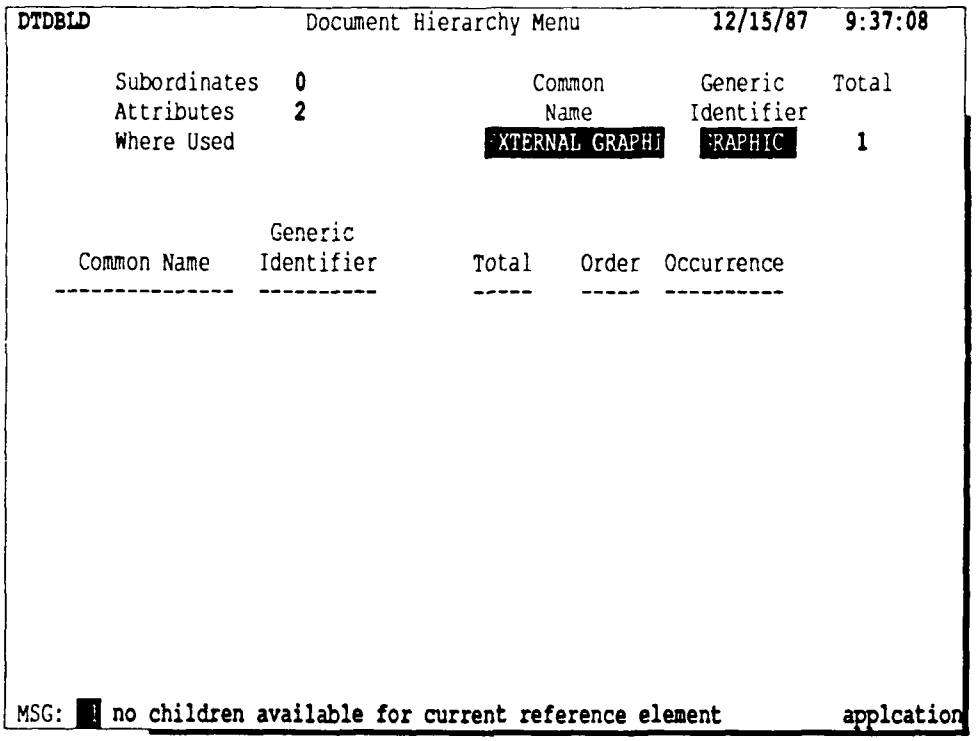

Figure 5-27 Document Hierarchy Menu Form after Attribute Test

Press the <QUIT> key again to return to the Document Hierarchy form as shown in Figure 5-28.

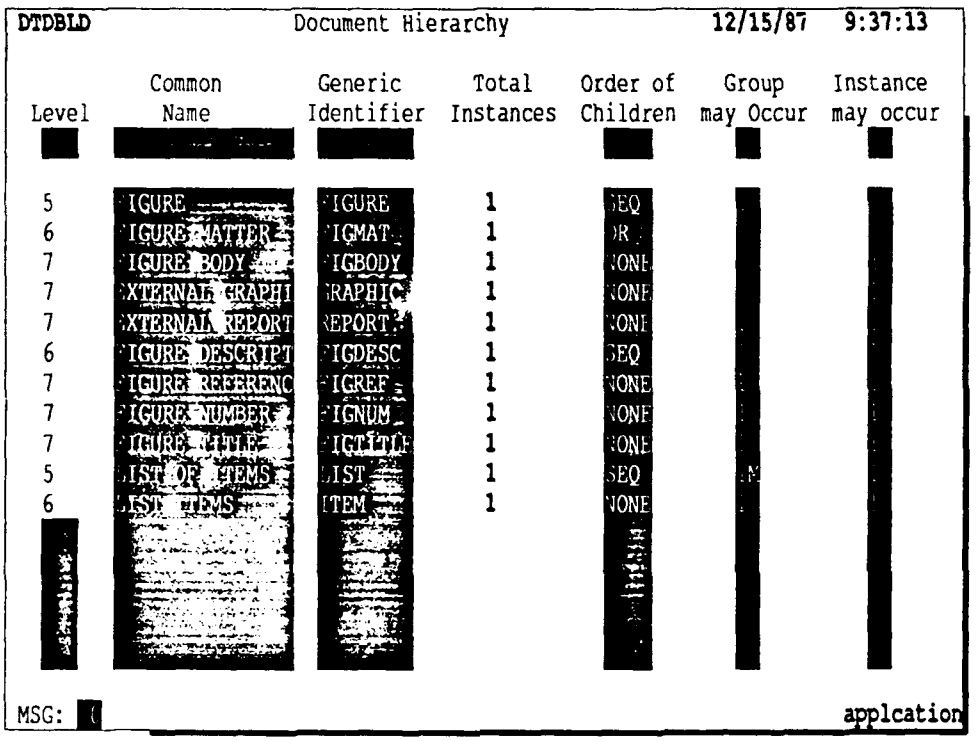

Figure 5-28 Document Hierarchy Form after Attribute Test

Now press the <PF6> key to save the Document Type Definition. A message asking whether it is okay to save the document type definition should appear as shown in the next Figure.

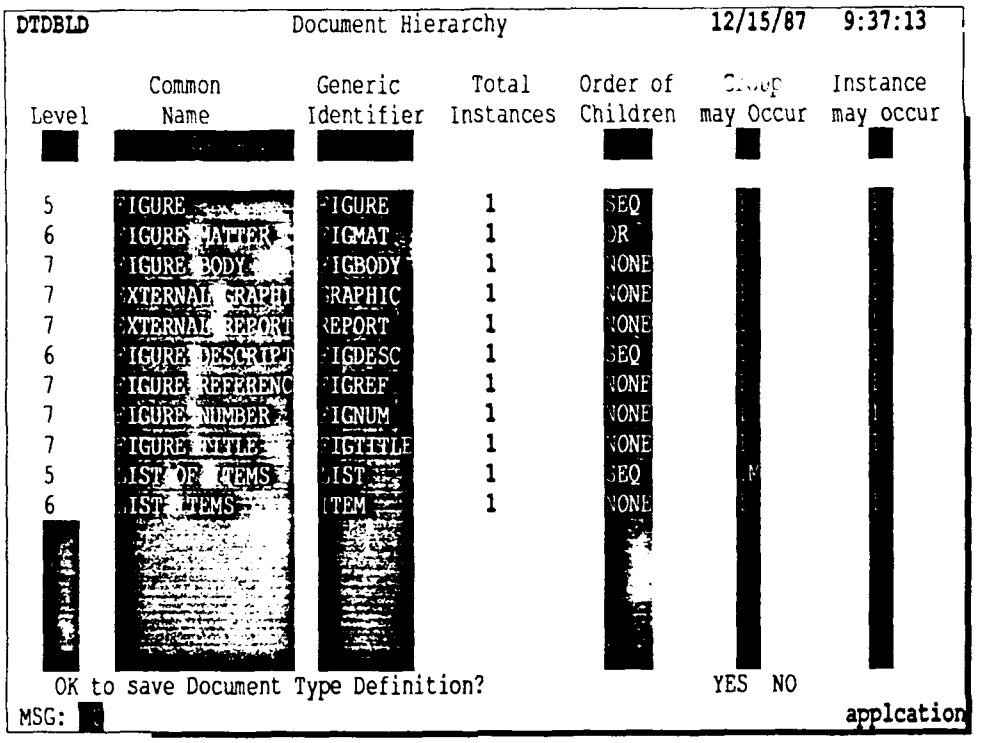

Figure 5-29 Document Hierachy Form after <PF6> Key Pressed

Press the <ENTER> key to save the Document Type Definition DTDBLD leaving a form looking as follows.

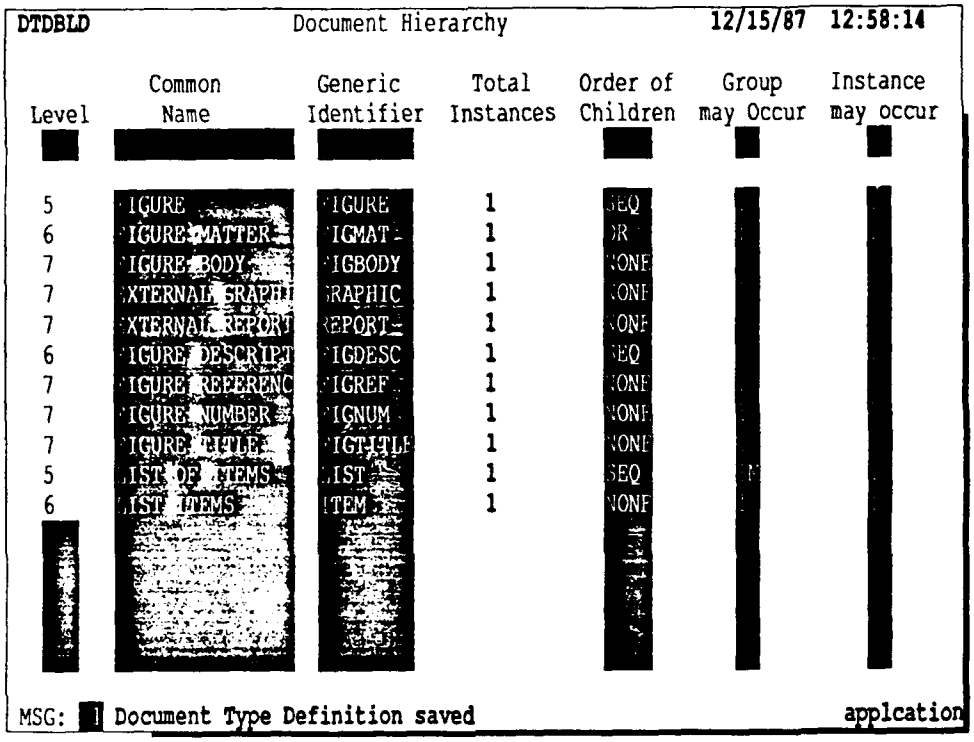

Figure 5-30 Completed Document Hierarchy Form after DTD Save

Now press the <QUIT> key to return to the DTD directory form as shown in the next Figure. Press <ENTER> when the application program asks whether to save the DTD.

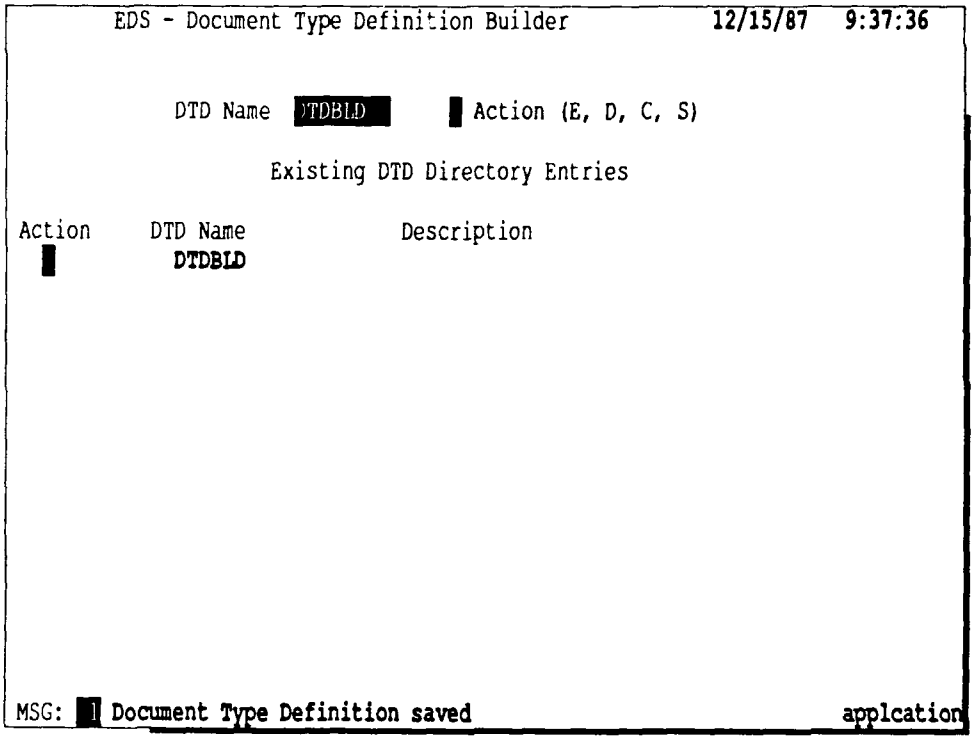

Figure 5-31 DTD Directory Form

The last part of the Unit Test Plan consists of testing the two DTD file maintenance commands of C - Copy and D - Delete. To test the Copy function blank the DTD Name and Action fields at the top of the screen and enter a "C" in the Action field for the DTD Name field. Press the <ENTER> key to obtain the COPY form as shown in the next Figure.

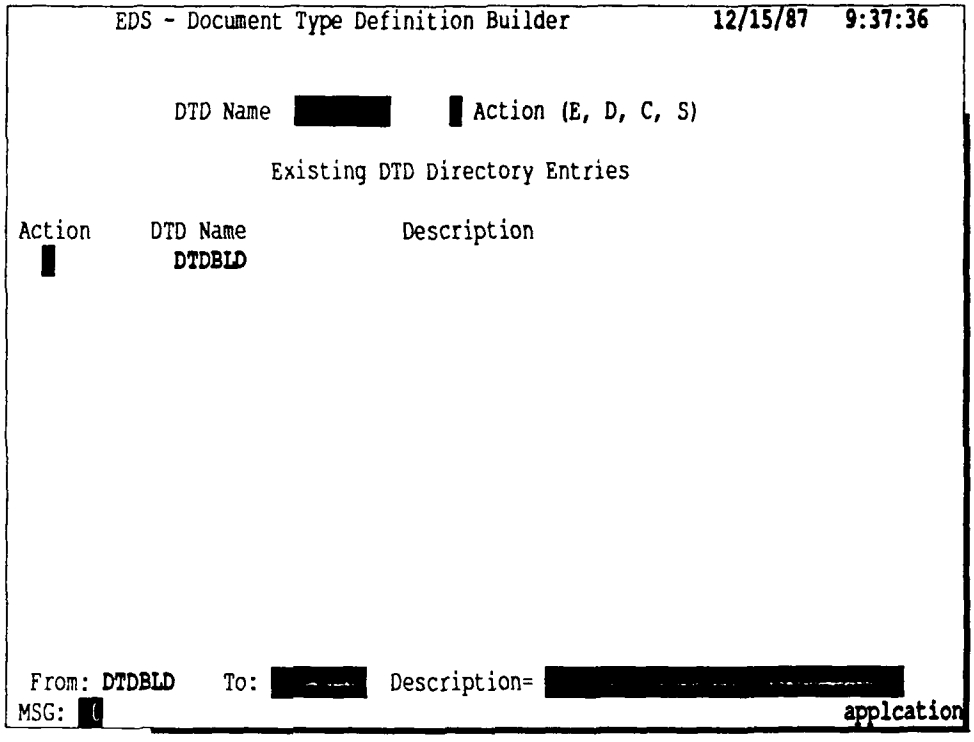

Figure 5-32 DTD Copy Test - Copy Form Template

Enter the data as shown in the next form for the COPY form.

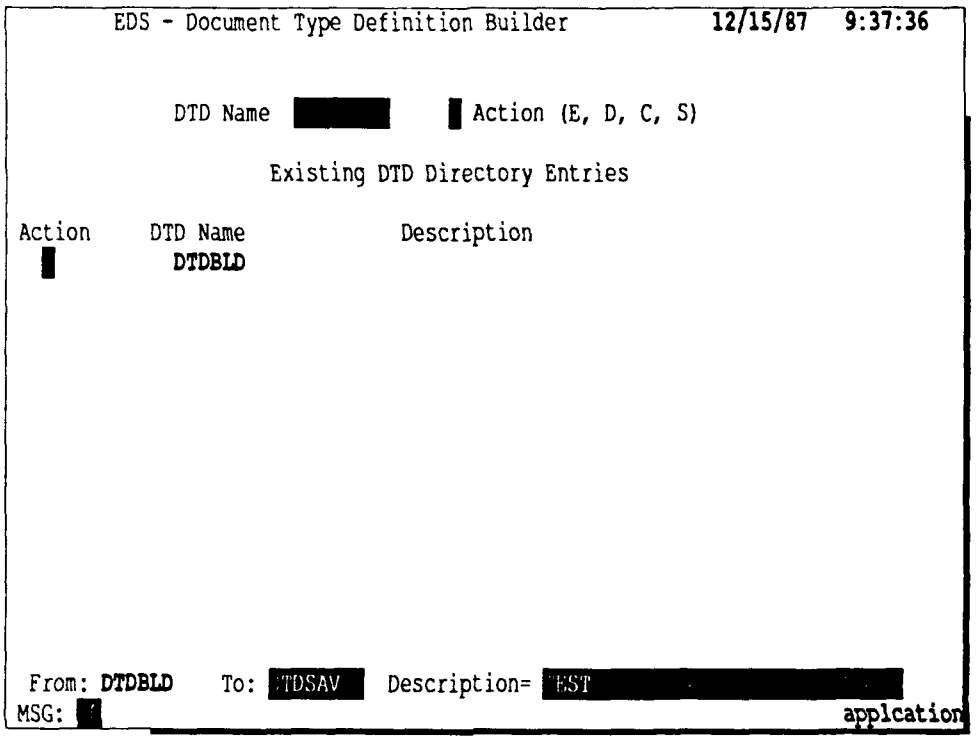

Figure 5-33 Input Data for Copy Test

Press the <ENTER> key to perform the COPY, resulting in a form as displayed in Figure 5-34.

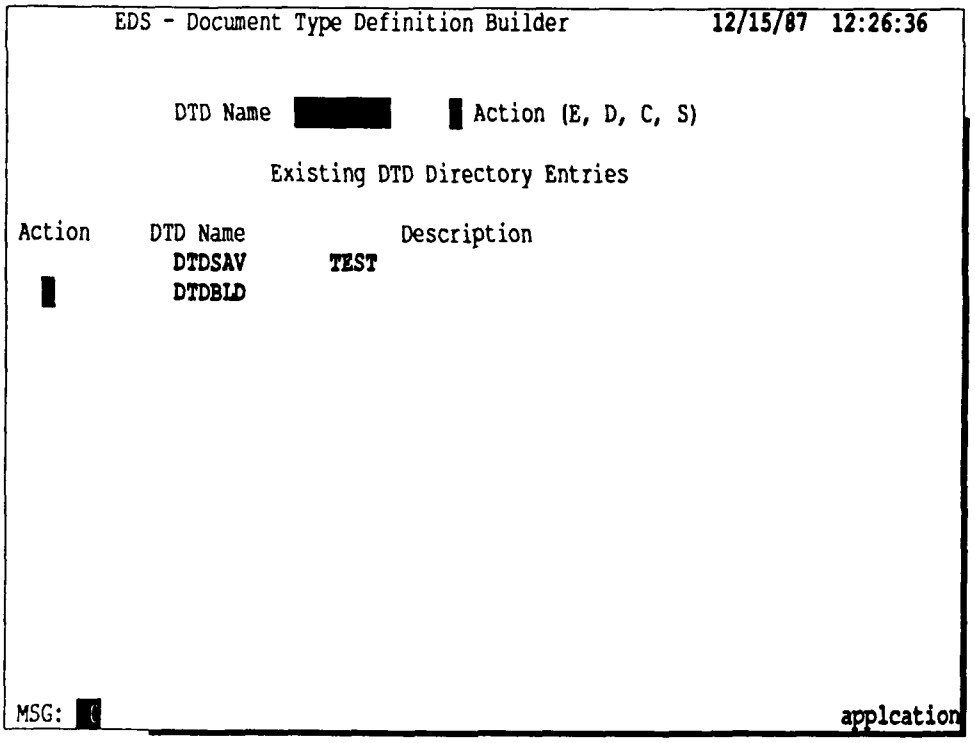

Figure 5-34 DTD Directory Form after Copy test

Now move the cursor to the Action field of the line containing the DTD Name of DTDSAV. Enter a D into the Action field and press the <ENTER> key to delete this DTD. A message should be displayed asking the tester to verify the delete operation as shown in the following form.

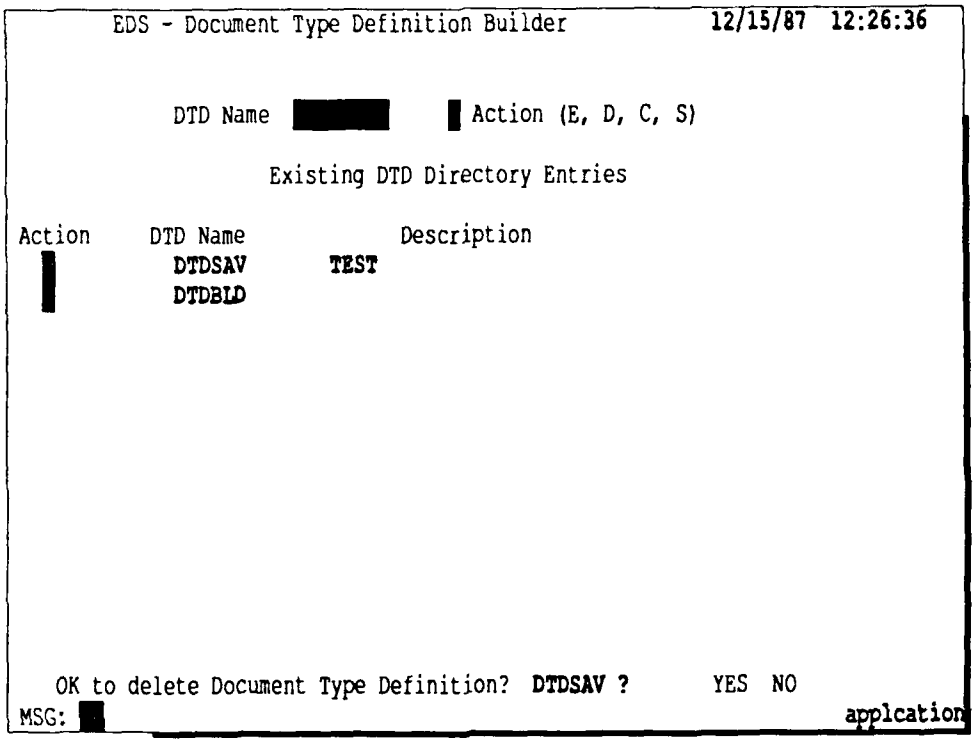

Figure 5-35 DTD Directory Form - Delete Test

Press the <ENTER> key to perform the Delete operation leaving a final form as shown in the next Figure.

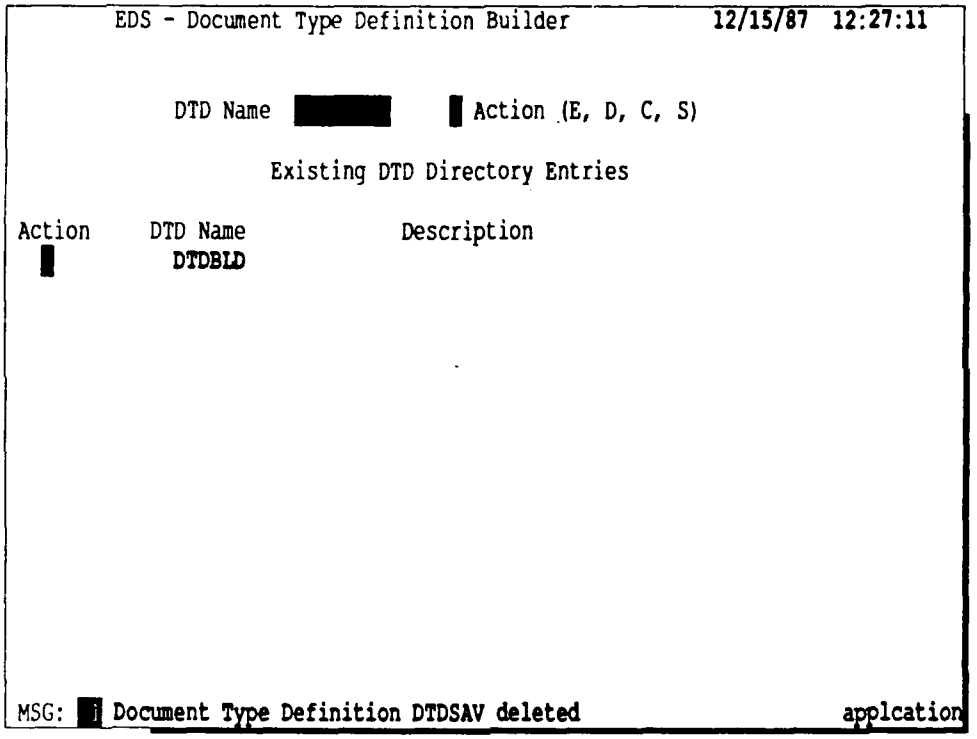

Figure 5-36 Final DTD Directory Form

The Unit Test Plan is now complete. Press the <QUIT> key to exit the DTDBLD application program. Once the application program has been exited, compare the output file DTDBLD.DTD with the file DTDBLD.SAV that is under IISS Configuration Management using a differences command. The two files should exactly match with exceptions for date/time for the Unit Test Plan to be successful.

U.S. Government Printing Office 648-12